Numerical Simulation of 3D, Complex, Turbulent Flows with Unsteady Coherent Structures: From Hydraulics to Cardiovascular Fluid Mechanics

A Thesis
Presented to
The Academic Faculty

by

Liang Ge

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Civil and Environmental Engineering
Georgia Institute of Technology
November 2004

Numerical Simulation of 3D, Complex, Turbulent Flows with Unsteady Coherent Structures: From Hydraulics to Cardiovascular Fluid Mechanics

Approved by:

Fotis Sotiropoulos, Advisor	Terry W. Sturm
Hermann Fritz	Donald Webster
Philip Roberts	Ajit P. Yoganathan

Date Approved: 16 November 2004

To my parents and wife.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Fotis Sotiropoulos, for his invaluable advice, support and encouragement, without which the completion of this dissertation would not have been possible. I would also thank the other committee members: Drs. Hermann Fritz, Philip Roberts, Terry Sturm, Donald Webster and Ajit Yoganathan for their valuable advices. Special thanks go to Dr. Sturm and his student Seung Oh Lee, Dr. Yoganathan and his student HwaLiang Leo for providing the experimental data and helpful suggestions.

I thank Feifei Pan, Casey Jones, Tahirih Lackey, Xiaodong Tian, Cristian Escauriaza, Janice Davice and other colleagues in the Water Resources group for their support and friendship during my study.

I would also express my gratitude to my parents, my sister and my parents-in-law for their understanding and support. I am greatly in debt to my wife, Bin, for her patience, love and encouragement. I would certainly have not been able to accomplish this without you.

This thesis was sponsored by National Science Foundation (NSF), National Institute of Health (NIH), and Georgia Department of Transportation (GA DOT).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Most engineering flows take place in geometrically complex, multi-connected domains and are dominated by unsteady coherent vortex shedding and turbulence. These attributes make the numerical simulation of such flows a rather challenging undertaking even for the most advanced computational fluid dynamics (CFD) methods available today. This thesis seeks to: a) develop an accurate, efficient, and versatile numerical solver that can accurately predict a broad range of complex engineering flows with unsteady statistical turbulence models; b) validate the numerical solver through comparisons with experimental measurements; and c) demonstrate the capabilities of the method by applying it to elucidate the physics of complex flows spanning a broad range of practical applications, from hydraulic engineering to bioengineering.

A new state-of-the-art CFD solver capable of simulating a broad range of complex engineering flows at real-life Reynolds numbers is developed. The method solves the three-dimensional incompressible unsteady Reynolds-averaged Navier-Stokes (URANS) equations closed with unsteady statistical turbulence models. Three such models are incorporated in the solver: the standard $k - \varepsilon$ model with wall functions, the Spalart-Allmaras model and the detached-eddy simulation (DES) model. The numerical solver employs domain decomposition with structured Chimera overset grids to handle complex, multi-connected geometries. The URANS equations are discretized in strong-conservation form using a second-order accurate, space-centered, finite volume method. A second-order accurate, dual- or pseudo-time stepping, artificial compressibility method is applied to integrate the discrete governing equations in time. During every physical time step, the equations are advanced in pseudo time with an efficient block-implicit, approximate factorization scheme. The turbulence model equations are discretized using flux-difference splitting upwind scheme for the convective terms and second-order accurate, central differencing for all other terms and advanced in time using the same dual-time step method used for the URANS equations.

To fully exploit the power of modern parallel computational facilities, the developed solver is parallelized using OpenMP.

The capabilities and versatility of the numerical method are demonstrated by applying it to simulate two widely different flow problems: a) flow past a geometrical complex array of multiple bridge piers mounted both on a natural river reach and on a flat bed experimental flume; and b) flow in mechanical, bileaflet, prosthetic heart valve with the leaflets fixed in the fully-open position. Overset grid systems with several millions of grid nodes are used and grid-refinement and other numerical dependency studies are carried out to explore the sensitivity of the computed solutions to various numerical parameters. For all simulated cases, large-scale unsteadiness appears naturally as a result of excited mean-flow instabilities and the computed mean flowfields are shown to be in good quantitative agreement with experimental measurements. By analyzing the instantaneous flowfields numerous novel insights into the physics of both flow cases are obtained and discussed extensively. The results of this thesis demonstrate the potential of the new method as a powerful simulation tool for a broad range of cross-disciplinary engineering flow problems and underscore the need for physics-based numerical modeling by integrating CFD with laboratory experimentation.

CHAPTER I

INTRODUCTION

1.1 Background

The advent of affordable supercomputers along with the proliferation of commercial and academic Computational Fluid Dynamics (CFD) software packages have contributed to the emergence of CFD as a powerful research and engineering design tool. Today CFD is widely used in a broad spectrum of engineering applications, spanning areas as diverse as river hydraulics and cardiovascular fluid mechanics. In spite of considerable progress, however, CFD methods capable of quantitatively accurate predictions of complex engineering flows at Reynolds numbers of practical relevance do not exist today. It is the major objective of this research to contribute toward the development and validation of such a method.

The main challenges in the accurate simulation of complex engineering flows stem from their enormous geometrical complexity and the rich flow physics such complex geometry tends to induce. Engineering flows take place in arbitrarily complex, multi-connected domains, which could involve moving boundaries and flexible bodies. Examples range from flows in natural river channels with complex hydraulic structures, such bridge foundations, intakes, hydropower installations, etc., to blood flow in artificial organs, prosthetic devices, and the cardiovascular system in general. The difficulty in modeling such flows is further exacerbated by the large disparity in scales between various geometrical features of interest and the dependence of the small-scale flow physics to phenomena occurring at the larger scale. In natural rivers, for example, the large-scale river bathymetry determines the approach flow hydraulics and, thus, greatly affects the local hydraulics in the vicinity of relatively small scale man–made hydraulic structures embedded within the river reach.

Engineering flows are often transitional or fully turbulent as they occur over a broad spectrum of Reynolds numbers, ranging from $Re = 10^3$ for cardiovascular flow applications to $10^6 - 10^7$ for hydraulic engineering applications. A common feature of such flows is

that they are all dominated by large-scale flow instabilities and unsteady coherent structures induced by the complex geometry. These coherent vortices typically scale with the geometry of the flow and their size is, thus, much larger than that of small-scale turbulent eddies. Yet coherent structures in such flows can contribute a significant percentage of the total turbulence kinetic energy and Reynolds stresses. Accurate simulation of such flows necessitates the use of efficient turbulence models that take advantage of the large disparity in flow scales to resolve the dynamics of the large-scale structures directly without having to resolve the entire spectrum of turbulent eddies, which would render the simulation of complex engineering flows impractical especially at high Reynolds number.

From a computational standpoint, the above challenges pose a host of numerical difficulties, which need to be addressed in a manner such that the resulting CFD method is not only accurate and reliable but also comprises an efficient, and thus useful, engineering simulation and analysis tool. To meet these challenges, a numerical algorithm must possess, among others, the following general attributes: 1) a versatile and easy-to-use grid generation approach that can easily lead to high quality grid system in arbitrarily complex geometries; 2) spatial and temporal discretization schemes with at least second-order accuracy; 3) practical turbulence models that can yield quantitatively accurate predictions with reasonable computational cost; and 4) efficient iterative schemes implemented in a code that is optimized to take full advantage of high–performance computational platforms and yield results within reasonable computational time. CFD algorithms which combine all of the aforementioned attributes do not exist today and their development is far from trivial as it hinges on the enormous complexities of real-life engineering flows.

The purpose of this research is to develop and validate a CFD framework that resolves the above challenges and to demonstrate the generality and versatility of the method by applying it to study the physics of two seemingly disjoint engineering problems: a) flows past complex bridge foundations in natural river reaches; and b) flows in prosthetic mechanical heart valves. Even though the context and Reynolds-number regime of these flow problems are rather diverse, they both involve very complex geometries and they are dominated by massive separation and large-scale, unsteady vortex shedding. Thus, their use as test cases

in this research will serve to gauge the capabilities of the numerical method and underscore its power as a general engineering flow simulation tool.

1.2 Literature Review

In this section we review previous work pertaining to the main computational challenges identified above. Namely, the issue of discretization of complex computational domains, the issue of turbulence modeling for complex flows, and the issue of efficient computer code implementation. The literature related to the specific flow problems we simulate in this work will be reviewed in separate subsequent chapters.

1.2.1 Numerical methods for incompressible flows

The time–dependent incompressible flow motion is described with

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\nabla p + \nu \Delta \mathbf{u} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{1}$$

Analytical solutions to the above equations only exit for several very simple cases. Solutions of almost all engineering flows have to be relied on numerical simulations. Numerical techniques required to obtain solutions of these equations include temporal and spatial discretization, solution of the discretized equations and boundary conditions. Every topic has been studied by a vast number of researchers and the state-of-the-art numerical techniques for incompressible flows are the result of these success studies. In the following sections, we subsequently discuss the issues of temporal discretization and spatial discretization related with incompressible flow simulation.

1.2.1.1 Temporal discretization

As seen in the governing equations of incompressible flow (Esq. 1), the governing equations of incompressible flows include three momentum conservation equations and one mass conservation (or continuity) equation. By given initial values of the pressure and velocity field, the velocity components can be advanced in time by integral the momentum equations. However, the continuity equation, which requires the time–dependent velocity field

to be divergence free thus serving as an incompressible constraint, does not contain any time–derivative of the pressure. Therefore, the governing equations lack any direct time–marching scheme for the pressure field. Schemes developed for compressible flows, where the flow variables (including density and velocities), can not be directly applied for incompressible flow simulation without any modification. Special considerations are necessary to obtain a pressure field that meets the incompressible constraints. Over the years, there are many different flavors of numerical methods developed to obtain time–accurate solution of the above governing equations. These methods can be roughly divided into two groups: pressure–based method and density based method.

1) **Pressure based methods**

As stated earlier, by given an initial condition of the pressure and velocity field, the momentum equations can be used to advance the velocity components in time. A naive, explicit integration can be described as

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = -RHS(U^n) \tag{2}$$

The solution at the new time step, U_i^{n+1}, does not satisfy the continuity equation, therefore dose not consist of the actual solution of the governing equations at the new time step. These velocity components have to be corrected to meet the incompressible constraint. The correction, δU_i, has an impact on the pressure field, thus can be used to obtain the correct pressure field at the new time step. Methods in this category include pressure Poisson method and pressure projection method.

- *Pressure Poisson Method*

This method was originally applied by Harlow and Welch (1965) in the MAC, Marker-and cell, method for the computation of free surface incompressible flows. In this method, by introducing intermediate velocity components which are taken from the momentum equations, the discretized continuity equation is transformed into a pressure Poisson equation.

The original pressure Poisson method reads as

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n - \nabla p^{n+1} + \nu \nabla \mathbf{u}^n \tag{3}$$

$$\Delta p^{n+1} = \nabla \cdot \left[\frac{1}{\Delta t}\mathbf{u}^n - (\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + \nu \nabla \mathbf{u}^n \right] \tag{4}$$

The above semi–discretization equations are advanced in time as follows: First, the pressure field at the new time step is obtained through the solution of the Poisson pressure equation. The computed pressure is then used to advance the momentum equations in time which yields the velocity solutions at the new time step. This new time step velocity field implicitly satisfies the continuity equation. The MAC method is first–order accurate in time. The explicit time–advancing nature imposes very strict limitation on stability and only very small time steps can be used.

- *Projection method (or fractional step method)*

Pressure projection method was independently developed by Chorin (1968) and Temam (1969). The projection method is based on the Hodge decomposition theorem, which states that any vector function $\mathbf{u}^*$ can be decomposed into a divergence free part $\mathbf{u}$ plus the gradient of the gradient of a scalar ϕ, i.e.

$$\mathbf{u}^* = \mathbf{u} + \nabla \phi \tag{5}$$

where $\nabla \cdot \mathbf{u} = 0$. The general projection method can be described by the following two–step predictor–correction procedure:

Step I: Prediction step

$$\frac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \Delta \mathbf{u} + \nabla \tilde{P} \tag{6}$$

where $\tilde{P}$ is the approximation of pressure field and $\mathbf{u}^*$ the intermediate velocity components. Since the $\tilde{P}$ does not equal to the actual pressure field at the new time step, P^{n+1}, the intermediate velocity field is not divergence–free.

Step II: Correction step

The next step in the projection method is to project these intermediate velocity fields into a divergence–free space, by using the pressure as a projection factor. As indicated by

the Hodge decomposition theorem, this can be done by

$$\mathbf{u}^* = \mathbf{u} + \Delta t \nabla \phi$$

Taken the divergence of the above equation will yield

$$\Delta t \Delta \phi = \nabla \cdot \mathbf{u}^* \tag{7}$$

The correct pressure field can be obtained

$$P^{n+1} = \tilde{P} + L(\phi) \tag{8}$$

where the correction function $L(\phi)$ depends on the selection of $\tilde{P}$.

Over the past few decades, many versions of projection methods have been developed by different researchers, by different choices of the pressure approximation in the prediction step. In the projection method developed by Kim and Moin (1985), the pressure is not used in the prediction step. This method is also referred as *pressure free projection method*. The scheme is written as follows:

Step I:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\frac{1}{2}(3C(\mathbf{u}^n) - C(\mathbf{u}^{n-1})) + \frac{1}{2}(\Delta \mathbf{u}^* + \Delta \mathbf{u}^n) \tag{9}$$

where $C(\mathbf{u})$ is the convective term.

Step II:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla \phi^{n+1} \tag{10}$$

with

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \tag{11}$$

Combining equations 10 and 11 together yields the Poisson equation for the auxiliary variable ϕ:

$$\Delta \phi = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \tag{12}$$

Although it's not necessary to have the values of pressure to advance the equations in time, they can be recovered as follows:

$$p = \phi - \frac{\Delta t}{2Re} \Delta \phi \tag{13}$$

As shown in the above equations, the convective terms are discretized with second–order–explicit Adams–Bashforth scheme and the viscous terms are discretized with the second–order–implicit Crank–Nicholson scheme. Such an implicit treatment of the viscous terms removes the numerical viscous stability restriction.

An important issue in this projection method is how to impose boundary condition for the intermediate velocity field. Since the pressure gradient does not appear in the prediction equation, the intermediate velocity u^* does not approximate u^{n+1}. As suggested by Kim and Moin (1985), in order to sustain the same accuracy as the discretization used for the velocity components, it is necessary to use the following boundary conditions for the intermediate velocity:

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla \phi^n \tag{14}$$

The scheme of van Kan (1986) uses $\tilde{P} = p^n$. The projection method can be written as follows:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\frac{1}{2}(C(\mathbf{u}^*) + C(\mathbf{u}^n)) + \frac{1}{2}(\Delta \mathbf{u}^* + \Delta \mathbf{u}^n) + \nabla p^n \tag{15}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{2}\nabla(p^{n+1} - p^n) \tag{16}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{17}$$

In the projection method developed by Bell et al. (1989), $\tilde{P}$ is selected as $p^{n-1/2}$. The method reads as follows:

Step I:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -C(\mathbf{u})^{n+\frac{1}{2}} - \nabla p^{n-1/2} + \frac{\nu}{2}(\Delta \mathbf{u}^* + \Delta \mathbf{u}^n) \tag{18}$$

where $C(u)^{n+1/2}$ is obtained with a Gudnov scheme.

Step II:

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla \phi^{n+1} \tag{19}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{20}$$

The pressure field is updated by

$$\nabla p^{n+1/2} = \nabla p^{n-1/2} + \Delta \phi^{n+1} \tag{21}$$

This scheme will lead to a first–order time accuracy for the pressure term and second–order time accuracy for the velocity components.

2) **Density based method**

This method is usually referred as *artificial compressibility* or *pseudo–compressibility* method in the literature. The method was first introduced by Chorin (1967). The idea of this approach is to recover the hyperbolic character of the compressible equations, by adding artificial– (or pseudo) compressible term into the continuity equation (this is where the name *pseudo–compressibility* came from). By this, the well–developed computational techniques for compressible flows can be directly applied on the simulations of incompressible flows. Although the original method was developed for steady flow calculations, it has been extended to unsteady flow simulations by performing local pseudo–time iterations for each physical time step, see Merkle and Athavale (1987) for 2D flows and Rogers et al. (1991) for 3D calculations. The detail idea can be expressed as:

$$\frac{\partial \mathbf{u}}{\partial \tau} + \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} \tag{22}$$

$$\frac{1}{\beta}\frac{\partial p}{\partial \tau} + \nabla \cdot \mathbf{u} = 0 \tag{23}$$

where τ is pseudo–time. By performing pseudo–time iterations in every physical time step and let all pseudo–time derivative terms ($\partial \mathbf{u}/\partial \tau$ and $\partial p/\partial \tau$) approaching to zero, the original unsteady governing equations are recovered.

An important issue in applying this pseudo–compressibility method is the iteration in pseudo–time. A fast and robust pseudo–time integration scheme is required to obtain a divergence–free solution at every physical time. The hyperbolic system above can be reorganized as

$$\frac{\partial Q}{\partial \tau} = RHS(Q) \tag{24}$$

where Q is the flow variables and RHS represents the residual. The above equation can be discretized into

$$\frac{Q^{k+1,n+1} - Q^{k,n+1}}{\Delta \tau} = \theta RHS(Q^{k+1,n+1}) + (1-\theta)RHS(Q^{k,n+1}) \tag{25}$$

where $0 \leq \theta \leq 1$. When $\theta = 0$, the above discretization is explicit, since the right hand side terms are all evaluated at the previous pseudo–time solution. If $\theta = 1$, the scheme is fully implicit while with other value of θ, mixed schemes are obtained.

Both explicit and implicit schemes are widely used, see Lin and Sotiropoulos (1997a) for explicit schemes and Rogers et al. (1991) and Yuan (2002) for implicit schemes. The advantages and disadvantages of both methods are well known in the literature. The explicit method is straightforward for implementation, has a less memory requirement and easy for extension to parallel computations. However, it suffers from the small time–step restriction imposed by stability requirement. In order to improve the convergence rate, methods like local time–stepping and residual smoothing (Hollanders et al., 1985) are developed and multi–grid methods are also very important in obtaining solutions with reasonable computational cost. On the other hand, implicit schemes, which needs the inversion of the coefficient matrix, requires much more memory per node and is more complex for implementation. However, since it has much larger stability range, larger time–step can be used for every single iteration thus leading to a faster convergence rate.

The semi–discretized form of the implicit scheme reads as

$$\frac{1}{\tau}\delta q = RHS(Q^{k,n+1}) + A\delta q + O(\delta q^2) \tag{26}$$

where $\delta q = Q^{k+1,n+1} - Q^{k,n+1}$ and A is the Jacobian matrix, $\partial RHS/\partial Q$. A prerequisite to solve the above equation is to obtain the inversion of Jacobian A, which is a huge sparse array. To directly inverse it requires extensive storage space and extremely long computational time. The system has to be solved approximately. Widely used methods include Alternative Directional Implicit (ADI) method (Beam and Warming, 1976), LU decomposition techniques (Jameson and Turkel, 1981; Caughey and Jameson, 2003), Newton–Krylov method (Ekici and Lyrintzis, 2003; Knoll and Keyes, 2004; Lowrie, 2004), etc.

1.2.1.2 Spatial discretization

Spatial discretization is a very important issue in solving the Navier–Stokes equations, especially the discretization of the convective terms and the pressure gradient. Schemes can be used to discretize the convective terms include upwind and central difference schemes.

Multidimensional upwind scheme can be constructed based on flux split method of Steger and Warming (1979) and Roe (1981), or Harten et al. (1983)'s TVD method. These methods are widely used in flow simulations for their stability. However, as shown in Mittal and Moin (1997), upwind like schemes are very dissipative and thus not suitable for physics–based numerical simulations, such as direct numerical simulation (DNS) or large–eddy simulations (LES). For this reason, non–dissipative central–difference schemes are always preferred in physics–based simulations.

Straightforward implementation of three–point central–difference scheme on a non–staggered grid may lead to non–physical pressure oscillations, the well–know odd–even decoupling problem. This problem can be effectively resolved by using the staggered grid method, which was originally proposed by Harlow and Welch (1965). In this method, the velocity components are stored on the grid cell edges while the pressure is stored on the cell center. A typical two dimensional staggered Cartesian grid is shown in Fig. 1. The x and y–directional momentum equations and the pressure equations are discretized on different control volumes. Such an arrangement removes the pressure oscillation from the calculation and does not require any special treatment on the pressure. Although originally specifically designed to remove pressure oscillation, it is found recently that the staggered grid arrangement had many attractive conservative properties. Lilly (1965) showed that, on top of the local mass and momentum conservation, Cartesian staggered grid also conserves circulation and kinetic energy if without viscosity. Zhang et al. (2002) developed unstructured staggered grid schemes that also conserves these properties. Because of these remarkable conservative properties of staggered grid, it is widely used in a lot of studies (Burton and Eaton, 2002; Hinatsu and Ferziger, 1991; Kirkpatrick et al., 2003).

The original staggered grid is implemented on the top of Cartesian grid system, where the Cartesian velocity components are normal to cell faces thus can be directly used to calculate the flux on cell faces. When this method is extend to three–dimensional curvilinear coordinate system, this orthogonality does not hold any more. In order to calculate the flux on cell faces with Cartesian velocity components, all three velocities have to be stored on every cell face, which leading to the definition of nine velocity component and solution

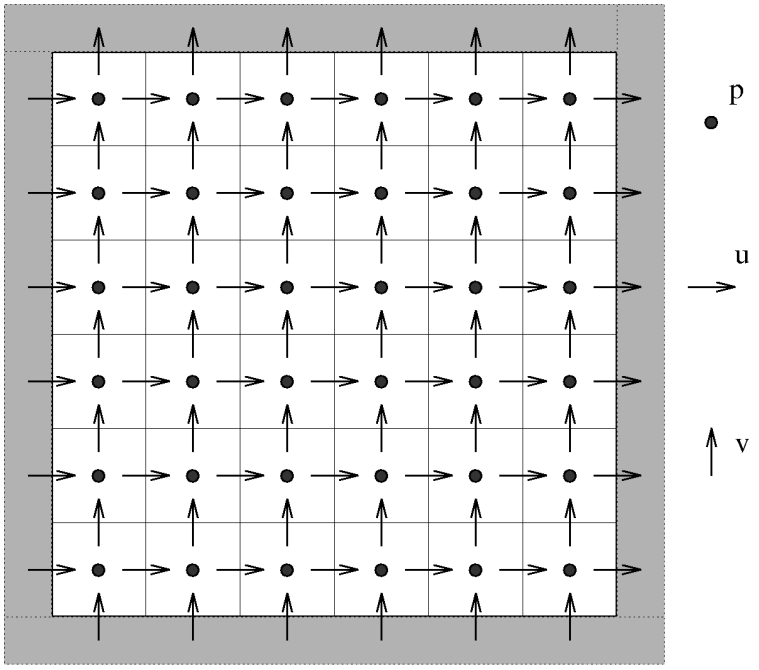


Figure 1: Variable storage arrangement of staggeredgrid grid

of nine momentum equations per control volume. This arrangement is inefficient thus should be avoided (Zang et al., 1994). A straightforward extension of the Cartesian grid arrangement is to store the contravariant velocity component on cell faces. By doing this, the whole governing equation system has to be transferred into the curvilinear system where contravariant and covariant velocity components are the system unknowns. This equation transfer procedure will lead to a very complicated system, which requires the evaluation of Christoeffl coefficients (Rosenfeld et al., 1991). Because of these difficulties, in complex three–dimensional simulations, non–staggered grid arrangement is preferred.

As shown in Sotiropoulos and Abdallah (1991), on the top of the non–staggered grid, it is impossible to yield a solution that is locally divergence free and without pressure oscillation from the central–difference scheme based discretized Navier–Stokes equations. The only remedy to this is to either apply some schemes that are inherently dissipative or to explicitly add artificial dissipation to the central difference. One of the most widely applied non–staggered grid methods was the Rhie and Chow interpolation scheme, originally proposed for steady flows on structured grid. The method was later extended to 3D unsteady

flow calculation by Zang et al. (1994) and was applied on the top of unstructured grid by Davidson (1996) and Choi (1999). The essence of Rhie and Chow interpolation method is to implicitly add a fourth–difference pressure term, which serves as artificial dissipation, to the continuity equation. In order to minimize the amount of the artificial dissipation introduced to the discretized continuity equation while maintaining the smoothness of the computed pressure field, Sotiropoulos and Abdallah (1991) proposed to explicitly add a fourth–difference artificial dissipation term in the discrete continuity equation. On convergence, the local divergence is equal to this explicitly added dissipation term thus the amount of artificial dissipation is controllable. Another way of introducing artificial dissipation to the discretized equations is to use artificial compressibility method to discretize the governing equations in time, thus the artificial dissipation schemes developed for compressible flows, such as Jameson and Turkel (1981), can be extended to the incompressible flows simulations. In Lin and Sotiropoulos (1997a), different forms of artificial dissipations, including scalar dissipation and matrix–valued dissipation, are compared.

1.2.2 Grid generation strategies

In general, there are two types of grid generation strategies: structured and un-structured grids. Structured grids, as indicated by the name, rely on structured coordinate systems and computational nodes are located at the intersection of the different families of coordinate lines. Structure grids use quadrilateral and hexahedral elements in 2D and 3D problems, respectively. In contrast, un–structured grids do not rely on a structured coordinate system and usually employ triangular and tetrahedral elements in 2D and 3D problems, respectively. Both approaches have their own strengths and weaknesses for CFD applications. The explicit organization of grid nodes in structured meshes leads to the efficient discretization of the governing equations of fluid flow and greatly facilitates the deployment of fast, implicit iteration solvers, such as approximate factorization schemes, and convergence acceleration methods, such as the multi–grid method, in the kernel of the numerical solver. Such features can largely enhance the efficiency of the numerical solver and result in fast turnaround time. Not imposing the often stringent requirement for a structured coordinate system, however,

can greatly facilitate and expedite grid generation in complex geometries, which is a very attractive feature of unstructured grid algorithms.

In spite of their attractive computational efficiency, the usefulness of structured grids is rather limited in real engineering flow problems due to the often sever restrictions posed on the grid generation method by the required structure of the grid elements. In fact, structured grids are well suited only for flow problems with simple boundary geometry, such as the flow around a single cylinder, as shown in Fig. 2. By adopting body-fitted, generalized, curvilinear coordinates, grids can be constructed for such simple geometries with elements aligned with the surface of the body, thus, allowing the easy implementation of boundary conditions and the efficient clustering of grid surfaces near the solid wall boundary layers. However, for more complex flow domains, such as the multi–cylinder flow problem shown in Fig. 3, using a single structured–grid system to discretize the entire domain is no–longer feasible. Such difficulties could be further exacerbated in flow problems involving geometrical features of interest at disparate spatial scales. As discussed above, for example, the local flow in the vicinity of bridge foundations is determined by the geometry of the foundation and the large–scale topography of the natural river reach within which the foundation is embedded. To accurately resolve the effect of the bridge foundation on the flow, very fine grid resolution would be required near the foundations. If a structured–grid method is adopted for such simulation a grid system with very large number of grid nodes would be required, which could result in an intractable computational problem even for the present-day, very powerful supercomputers. There are two options to remedy these difficulties: 1) unstructured grids; and 2) domain decomposition with locally structured grids.

Unstructured grid methods originally emerged as an alternative to structured grid for complex geometries simulations. These methods use tetrahedral or hexahedral or mixed type of elements. The elements are allowed to be arbitrarily connected with neighboring elements. This not only provides extra feasibility for discretizing complex geometries but also allows for the relatively easy implementation of locally adaptive algorithms to simulate

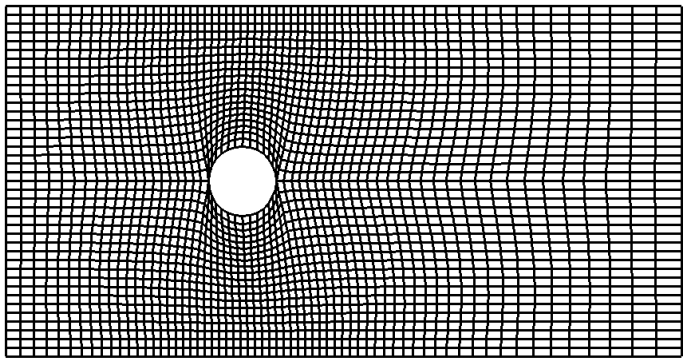


Figure 2: Structured grid around a single cylinder

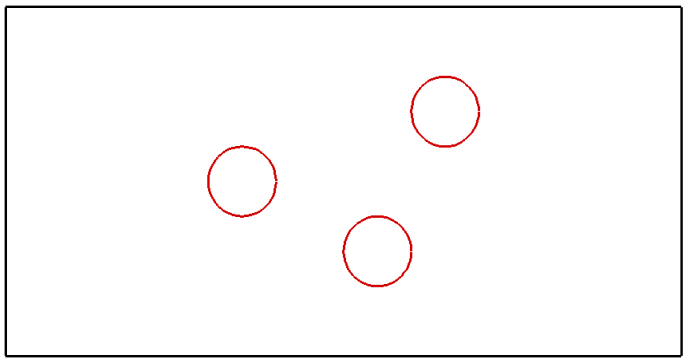


Figure 3: Multi-cylinders problem. The flow domain is very difficult to be discretized with a single block structured grid system

problems with multiple geometrical scales. Because of these attractive features, unstructured gird methods are actively studied by many researchers. Researches are focused on automated grid generation in complex geometries (Marcum and Weatherill, 1995; Owen and Saigal, 2000; Marcum, 2001), spatial discretization (Kobayashi et al., 1999; Zhao and Zhang, 2000; Zhao et al., 2002; Zhang et al., 2002; Darwish and Moukalled, 2003; Mahesh et al., 2004), and solutions techniques (Kim and Choi, 2000; Moinier et al., 2002; Ollivier-Gooch, 2003; Sorensen et al., 2003; Lambropoulos et al., 2004).

The unstructured grids' general complex geometry handling capability comes with huge

cost penalties. Unstructured grid methods require much more memory (to save grid connectivity information) than structured ones. Moreover, because of lack of grid structure, certain PDE solution acceleration techniques, such as ADI methods, multi–grid techniques, are very difficult or even impossible to implement on the top of unstructured grid systems and the solution of PDEs has to be relied either on explicit time integration schemes (e.g., Runge–Kutta method) or Newton–based implicit schemes (Brown and Saad, 1990; Blanco and Zingg, 1998; Nemec and Zingg, 2002). The explicit time integration schemes suffer from the strict time–step requirement and require much more iteration steps to obtain converged solution. On the other hand, the Newton like schemes benefit from the fact that all equations are advanced in an implicit method thus allows for the usage of large time–step. However, these methods require the inversion of a very large sparse matrix, either directly or approximately. Such inversions are very expensive, both on the sense of memory requirement and computational time cost. Therefore, solvers based on unstructured grids are usually much more expensive in the sense of computational time cost. Another cost to use unstructured grid is the loss of accuracy. Straight forward implementation of second order interpolation schemes of the flow variables will lead to first order approximation of the flux on control volume faces on an arbitrary mesh (Hirsch, 1999). Great efforts have been paid on developing fully second–order accurate schemes that is independent of grid, success has only been reported and applied on 2D cases (Kobayashi et al., 1999; Abgrall, 2001; Abgrall and Mezine, 2003). In 3D simulations, second–order accuracy can only be reached with uniform unstructured grid (Zhang et al., 2002; Mahesh et al., 2004), which is impossible on complex geometry simulations. When it comes to high–Reynolds turbulent flow simulations, the thin boundary layer usually requires mesh spacing several orders of magnitude smaller in the direction normal to the wall than in the streamwise direction, resulting in large cell aspect–ratios in these region. Such large aspect–ration cells may decrease the accuracy of solution (Babushka and Aziz, 1976) and should be avoid. The current industry applications typically avoid this problem with hybrid grid, which combines structured–grid near boundary with unstructured grid in the core flow region (Khawaja and Kallinderis, 2000; Douglass et al., 2002; Athanasiadis and Deconinck, 2003). Due to all the above limitations,

unstructured grid is not ready for large–scale viscous flow simulation and huge development is required before it reaches such a goal.

An alternative to unstructured grids is the use of multi–block approach with locally structured grids. In this hybrid methodology, which attempts to combine the benefits of fully structured and unstructured approaches, a multi-connected computational domain is first divided into a set of several smaller and geometrically simple subdomains (or blocks). The shape of each block is selected such that it can be easily discretized by a locally structured grid, whose coordinates could in general be independent of those used in adjacent blocks. The idea of splitting a complex domain into small blocks is called domain decomposition. The domain decomposition method (DDM) was first proposed by Schwartz (1869). Generally there are two groups of DDM: overlapping and non–overlapping (or patched) DDM. The difference between these two groups lies in the arrangement of blocks. In the former approach neighboring sub–domains are allowed to arbitrarily overlap, while in the latter approach blocks must exactly match with the others along their common interface. The non–overlapping DDM method, which is also referred as patch grid methods, have been applied in many complex flow simulations, such as flow around a train (Khier et al., 2000), flow around model airplanes (Takanashi and Takemoto, 1995), and unsteady flows in a centrifugal compressor stage (Koumoutsos et al., 2000) etc. Since the blocks sharing the same interfaces on the boundaries, conservation schemes that guarantee flux conservation on interfaces are relatively easier to develop and implement (Wright and Shyy, 1993). However, the interface mapping requirement of non–overlapping DDM poses some extra constraints on grid generation, making their usefulness in extremely complex flow simulations rather limited.

The recent implementation of overlapping DDM in CFD was proposed by Steger and Benek (1987) using the so-called Chimera or overset grid approach. To illustrate the idea of a Chimera grid, let us refer to the simplified problem shown in Fig. 4. The boundaries of the physical domain are shown as solid lines. The original domain can be divided into two domains, a circular region and a rectangular region, by supplying auxiliary lines shown as dashed–line (which will be referred to as the *grid interface*) on the figure. Structured

grids can be generated for each of the two subdomains separately, i.e. without any regard to the mesh topology used in the other region. Solving the governing flow equations in such a composite domain requires very careful treatment of the flow variables at the subdomain interfaces to ensure that flow structures generated in one subdomain can cross without spurious distortions and affect the flow in adjacent subdomains. Typically, the flow variables at the interface are obtained via an iterative procedure employing some kind of an interface interpolation scheme, which communicates information from one subdomain to the others.

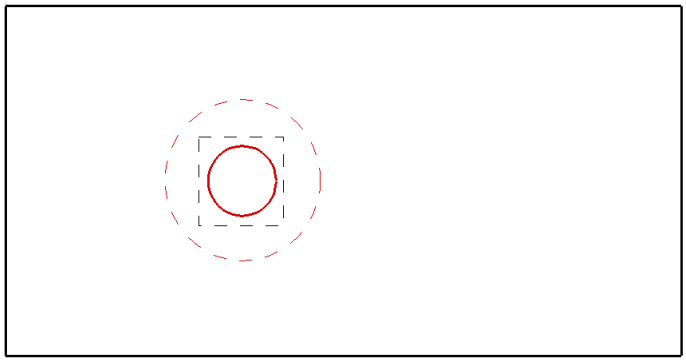


Figure 4: Chimera grid method for flow around a single cylinder

Specification of boundary conditions at the grid interfaces is the most important numerical issue in algorithms relying on the Chimera approach. A good treatment of grid interface boundary conditions should reproduce the flow physics as described by the original governing equations on the same physical domain (without domain decomposition), all flow structures generated in one block to cross without distortion into neighboring blocks, ensure numerical stability, and eliminate spurious oscillations of the flow variables across the interfaces. The grid interface treatment schemes can be generally classified as conservative and non–conservative methods, depending on whether they ensure or not conservation of certain physical quantities or fluxes across the interface. The most straightforward scheme is to use linear interpolation. The variables at grid interfaces are specified through linear

interpolation based on the relative position of the grid nodes on interface and the background grid cell. This implementation leads to a non–conservative method. It is widely used because of its simplicity and efficiency on implementation (Benek et al., 1983; Chesshire and Henshaw, 1990). This implementation, however, does not guarantee global conservation of mass and momentum fluxes, i.e., the integration of mass and momentum fluxes on the composite grid interface does not equal to zero. Lack of global conservation could be detrimental to the accuracy and smoothness of the computed solution, especially for incompressible flow simulations where global mass conservation is a necessary condition for the existence and uniqueness of a smooth pressure field (Hubbard and Chen, 1990). As stated earlier, global conservation can be ensured through flux interpolation on the shared grid interface of patched multi–block structured grids, such as schemes developed by Rai (1986) for compressible flow and by Wright and Shyy (1993) for incompressible flows. However, the interface mapping requirement of non–overlapping DDM poses some extra constraints on grid generation, making their usefulness in extremely complex flow simulations rather limited. The overlapping block arrangement of Chimera grid methods removes the requirement of grid matching on interfaces thus largely improves the complex geometry handling capability. Meanwhile, the lack of common grid interfaces makes it extremely difficult to enforce global conservation, especially for 3D.

To date, there is no clear evidence to support the relative superiority of conservative and non-conservative interface interpolation schemes. As argued in Meakin (1995), the key issue of the Chimera approach is that of grid resolution in the vicinity of the interface rather than the conservative character of the interface interpolation algorithm. An interface interpolation scheme that attempted to balance the requirement for conservation at the interface and computational simplicity and expedience in 3D simulations was proposed by Tang (2001), who proposed a new grid interface scheme based on enforcing a second-order accurate discrete approximation of global mass conservation at each overset grid interface. This scheme was shown to enhance the overall efficiency of the numerical algorithm as well as the smoothness of the computed solution across interfaces as compared to the standard trilinear approach, especially when the grid spacing of adjacent subdomains is discontinuous.

The Chimera grid method is desirable in CFD simulation of engineering flows in many aspects. First, this method allows the arbitrary overlapping between subdomains and poses little restrictions on the block interfaces, thus, leading to the highest feasibility in the process of domain–decomposition. Changes of the computational domain—obtained, say, by adding or deleting solid obstacles or changing the shape of existing ones—can be easily accounted for by altering, adding or removing grid components to the grid system, while leaving the overall or background grid system untouched. Such a feature is especially useful for multi–connected flow domains.

Second, the grid arrangement in a specific block is not directly related to the other blocks. Therefore, grids in each block in the system can be locally optimized to obtain the best result. For example, in turbulent flow simulation, very fine grids near solid walls are required. By carefully arrangement of blocks in overlapping DDM, grids can be allowed to be clustered to the regions where it is necessary. Another attractive feature of overlapping DDM is it provides a natural level of parallelism for execution on modern parallel computers.

Finally, the ability of the overset grid approach to handle arbitrarily overlapping grids with discontinuous grid spacing in adjacent subdomains facilitates the implementation of grid embedding strategies. Grid embedding is ideally suited for handling geometries with large disparity in spatial scales. For the previously discussed example of bridge foundations embedded in a river reach, for example, a background grid fine enough to resolve the large scale topographical features of the reach can be used to simulate the approach flow hydraulics in the vicinity of the structures. The local hydraulics in the vicinity of the structures can then be accurately modeled by embedding successively finer grids within the background mesh to essentially zoom in spatial scales comparable to those of the unsteady, coherent vortices shed by the structures.

Because of overset grid's superior capability in handling complex geometry flow problems, this method has be widely studied and applied since its appearance. The capability of overset grid in handling complex geometries is well demonstrated in a number of researches. Rogers et al. (2001) used the overset grid method to calculate the viscous flow around a complete Boeing 777-200 aircraft configured for landing. The geometry of a 4.2%

scale model aircraft used in wind tunnel experiments was used for simulation. All major components, including the fuelsage, the main wing, the inboard and outboard leading–edge slats, the Krueger slat, the inboard and outboard flaps, the flaperon, the engine strut and the vertical tail, are considered in the simulations. Overset grids are generated with PEGASUS, an automated overset grid generation solver. A grid system with overall grid nodes of 22 million is used for this extremely complex geometry. Simulations are conducted with OVERFLOW, a parallel multi–block numerical solver. The convective terms are discretized with third–order Roe upwind differencing scheme and the turbulence is modeled with Spalart–Allmaras turbulence model (Spalart and Allmaras, 1994). Steady state solutions were obtained from the simulations. Aoki (2001) developed a numerical solver, which combines the Interpolated Differential Operator (IDO) scheme, Cut Cell technique and Chimera grid method, to simulate the falling process of papers. The papers are treated as rigid and a high resolution structured grid block following the motion of paper is designed to accurately reproduce the flow physics in the vicinity of the paper. The simulation reproduced the fluttering trajectory of paper falling and shows the mechanism controlling this process. Pan and Damodaran (2002) developed unsteady incompressible flow solver based on Chimera overset grid method. The time–dependent Navier–Stokes equations are solved by using a pressure–projection method. The convective terms are discretized with second–order accurate Gudnov method. The method is applied to study the flows around a sphere (Pan and Damodaran, 2002), flows around an oscillating cylinder and flows around multi–body underwater robotic vehicles (URV). Chimera overset grid method is also applied to investigate the 3D unsteady turbulence flows. Chen and Liu (1999, 2000) investigated the unsteady turbulent flow induced by ship berthing. Hedges et al. (2002) studied the turbulent flow field around a simplified landing gear by detached eddy simulation (DES) and unsteady RANS (URANS) approaches. As shown in these examples, Chimera grid method is a suitable choice for 3D unsteady complex flow simulations.

1.2.3 Turbulence modeling

Most flows of engineering interest are either transitional or fully turbulent and, thus, the issue of turbulence modeling is a key consideration when developing predictive CFD algorithms for complex flows. The challenge in modeling turbulent flows stems from the very nature of turbulence, which is highly three-dimensional and unsteady with rich chaotic dynamics and is characterized by a broad range of spatial and temporal scales (usually referred to as turbulent eddies). In a turbulent flow the size of the eddy motions spans a very wide spectrum, ranging from the smallest scale, l, which is defined as the scale at which the viscous effects are dominant, to the largest scale, L, whose magnitude is of the same order of the length scale of flow domain. The difference between these two scales is estimated, through dimensional analysis, to be proportional to $Re^{4/3}$ (Piomelli and Balaras, 2002). This estimate shows that even for flows at moderate Re, the disparity in scales is so large that it imposes a great challenge in numerical simulations and necessitates the adoption of some kind of modeling strategy for simulating engineering turbulent flows.

Turbulent flows of incompressible fluids are governed by the three-dimensional, time-dependent continuity and Navier–Stokes equations as shown in Eq. (1). The brute-force approach for simulating turbulent flows is to solve the above equations via the so-called direct numerical simulation (DNS) approach. In this approach, the time-dependent governing equations are solved directly by using spatial and temporal resolution fine enough to resolve all turbulent eddies, from the geometry-dependent largest scales to the viscosity-dependent scales, which dissipate the energy of the turbulent motion into heat. Assuming that a numerical method free of errors is employed, DNS can provide the most accurate description of any turbulent flow. This, however, can only be accomplished at a great computational cost as the computational mesh needs to be fine enough to resolve the smallest scales (the Kolmogorov scale) of turbulence. As shown through the previous scale estimate, the total number of grid nodes necessary to represent all scales of motion in a turbulent flow is of the order of $Re^{9/4}$. Because of this power-law dependence of grid size on Re, DNS is not a viable engineering simulation tool in high Re engineering flows even on today's fastest supercomputers. For that reason DNS is almost exclusively used today for

simulations of low-Re turbulent flows in relatively simple geometries to obtain accurate and complete descriptions of turbulent flow fields, which can be used to develop, refine and test the predictive capabilities of various turbulence modeling approaches (Moin and Mahesh, 1998).

Turbulence modeling techniques constitute the only practical alternative for simulating complex engineering flows at Reynolds numbers of practical interest. In general, all turbulence modeling strategies employ some kind of decomposition of instantaneous fluctuating flow variables (velocities, pressure, scalar concentration) into a filtered or resolved component and a fluctuated or unresolved component, which needs to be modeled. The scale of the resolved component is usually much larger than that of the unresolved component and can, thus, be directly resolved with relatively less computational cost. The fluctuating part of the turbulence flow field, however, remains unknown and its effect on the resolved flow field needs to be taken into account by introducing a turbulence model.

Currently, there are two widely used filtering approaches: one is to filter the flow field in space and the other is in time. These two different filter processes lead to two different categories of widely used turbulence modeling techniques: large–eddy simulations (LES) and Reynolds–averaged numerical simulations (RANS). Although the mathematical and physical framework of these two different approaches are quite different, they both share a common goal insofar as they both seek to model the unresolved fluctuating field in terms of resolved flow quantities.

1.2.3.1 Reynolds–averaging and the Reynolds-averaged Navier–Stokes (RANS)

RANS turbulence models constitute the most popular turbulence modeling technique for complex, engineering flow simulations, presumably due to their relative simplicity and overall computational expedience. All RANS models are based on the Reynolds-averaged Navier–Stokes equations, which are obtained by first decomposing instantaneous flow quantities as a sum of a deterministic and a random part and then either time– or ensemble–averaging the instantaneous continuity and Navier–Stokes equations. In this section we outline the basic principles of the averaging process, which lead to the derivation of the

RANS equations, summarize the various types of turbulence closure models, and review past work both in steady and unsteady RANS modeling with emphasis on complex 3D shear flows.

At any given point $\overrightarrow{x} = (x_1, x_2, x_3)$ in the flowfield an instantaneous flow quantity, say the velocity field u_i, can be decomposed by adopting the following general triple decomposition proposed by (Hussain and Reynolds, 1972):

$$u_i\left(t, \overrightarrow{x}\right) = \overline{u_i}\left(\overrightarrow{x}\right) + \widehat{u_i}\left(t, \overrightarrow{x}\right) + u_i'\left(t, \overrightarrow{x}\right) \tag{27}$$

where $\overline{u_i}$ is the mean value defined as:

$$\overline{u_i}(\overrightarrow{x}) = \lim_{T\to\infty} \frac{1}{T} \int_0^T u_i\left(t, \overrightarrow{x}\right) dt \tag{28}$$

$\widehat{u_i}$ is the deterministic or coherent contribution (see subsequent discussion) and u_i' is a random, turbulent fluctuation.

The deterministic component of the above decomposition is assumed to have a characteristic time scale, which is much greater than the greatest time scale of the turbulent fluctuations. In other words, the decomposition given by Eq. (27) assumes that there is a clear spectral gap between the deterministic and turbulent fluctuations. Such low frequency component could be either due to a forcing imposed onto the flow or due to naturally excited, large scale, flow instabilities. In the former category, for example, one would classify flows encountered in turbomachinery applications, where the rotational frequency of the rotor imposes an unsteady forcing, as well as cardiovascular flows, where the forcing is imposed by the cardiac cycle. In such a case, $\widehat{u_i}$ at any instant in time can be obtained by ensemble-averaging the results of many experiments at the same moment during the phase of the deterministic unsteady forcing and subtracting the mean. Examples of flows in the latter category are flows involving turbulent shear layers, which are dominated by large-scale coherent structures due to the Kelvin-Helmholtz instability (shedding behind solid obstacles, mixing layers, etc.). Once again the concept of phase-averaging could also be employed to interpret $\widehat{u_i}$. Alternatively, one could also obtain $\widehat{u_i}$ by time-averaging the instantaneous quantity over a finite-size temporal window whose size is much smaller than

the time-scale τ_d of the deterministic component and much greater than the largest period τ_t of the incoherent, turbulent fluctuations:

$$\widehat{u_i}(t, \overrightarrow{x}) = \frac{1}{\tau} \int_t^{t+\tau} u_i \left(t', \overrightarrow{x}\right) dt' - \overline{u_i}\left(\overrightarrow{x}\right)$$

where $\tau_t << \tau << \tau_d$.

1) **Steady RANS**

Steady RANS modeling is of course meaningless for flows involving forced, low frequency unsteadiness but it has been widely used in the past and continues to be used for all other flows. In this context the deterministic and turbulent fluctuations are lamped together into a single random fluctuation term and the instantaneous velocity field is decomposed as follows:

$$u_i\left(t, \overrightarrow{x}\right) = U_i\left(\overrightarrow{x}\right) + U_i'\left(t, \overrightarrow{x}\right)$$

where:

$$U_i = \overline{u_i}$$

and

$$U_i' = \widehat{u_i}\left(t, \overrightarrow{x}\right) + u_i'\left(t, \overrightarrow{x}\right)$$

By substituting the above decomposition into the instantaneous Navier-Stokes equations and applying time-averaging we obtain the steady RANS equations:

$$\begin{aligned} \frac{\partial U_i}{\partial x_i} &= 0 \\ \frac{\partial U_i U_j}{\partial x_j} &= -\frac{\partial p}{\partial x_i} + \nu \nabla^2 U_i + \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_j} \end{aligned} \tag{29}$$

where τ_{ij} is the symmetric Reynolds–stress tensor:

$$\tau_{ij} = -\rho \overline{U_i' U_j'}$$

which introduces six more unknowns into the governing equations as a result of the averaging process. Closing the RANS equations, therefore, requires a turbulence closure model for the six Reynolds stresses. An attempt to derive equations for the Reynolds stresses by further manipulating the Navier–Stokes equations inevitably leads to equations with higher-order turbulent correlations, such as triple fluctuating velocity products, products of velocity and pressure fluctuations, etc. This goes on ad infinitum as any attempt to derive transport equations for these new terms continues to introduce higher order terms. The inability to close the RANS equations is known as the turbulence closure problem, which can only be solved by modeling. A variety of turbulence models have been proposed in the literature, which, depending on the general approach adopted to model the Reynolds stress terms, are classified as isotropic eddy viscosity models, non-linear eddy-viscosity models, and Reynolds-stress models. The reader is referred to the reviews by Sotiropoulos (2000) for a more detailed overview of various closure models for the steady RANS equations. For brevity, here we only present the equations of two well known isotropic models, which we will subsequently employ in this research: the standard $k-\varepsilon$ model, which is a two-equation, isotropic eddy viscosity model; and the Spalart-Allmaras eddy-viscosity model, which is a one-equation isotropic model.

All isotropic models rely on the so-called Boussinesq approximation, which postulates that the Reynolds stresses depend linearly on the mean rate of strain:

$$\tau_{ij} = -\frac{1}{3}\overline{U_k' U_k'}\delta_{ij} + \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \tag{30}$$

where δ_{ij} is Kronecker's delta (the unit tensor) and ν_t is the turbulent eddy viscosity, which unlike its molecular counterpart that inspired the above model is a property of the flow and not of the fluid. The approach adopted to calculate ν_t is what differentiates the various isotropic models. The most widely applied model is the standard $k-\varepsilon$ model (Jones and Launder, 1972), in which the eddy–viscosity is given by

$$\nu_t = C_\nu k^2/\varepsilon \tag{31}$$

where k is the turbulence kinetic energy and ε is its rate of dissipation and are obtained by

solving the following transport equations:

$$\begin{aligned}\frac{\partial k}{\partial t}+\frac{\partial U_j k}{\partial x_j}&=\frac{1}{\rho}\tau_{ij}\frac{\partial U_i}{\partial x_j}-\varepsilon+\frac{\partial}{\partial x_j}\left[(\nu+\nu_t/\sigma_k)\frac{\partial k}{\partial x_j}\right]\\ \frac{\partial \varepsilon}{\partial t}+\frac{\partial U_j \varepsilon}{\partial x_j}&=C_{\varepsilon 1}\frac{\varepsilon}{k}\frac{1}{\rho}\tau_{ij}\frac{\partial U_i}{\partial x_j}-C_{\varepsilon 2}\frac{\varepsilon^2}{k}+\frac{\partial}{\partial x_j}\left[(\nu+\nu_t/\sigma_\varepsilon)\frac{\partial \varepsilon}{\partial x_j}\right]\end{aligned} \tag{32}$$

where the various constants assume their standard values: $C_\mu = 0.09$, $c_{\epsilon 1} = 1.44$, $c_{\epsilon 2} = 1.92$, $\sigma_k = 1.0$ and $\sigma_\epsilon = 1.3$. There are also many other flavors of RANS models available.

The Spalart Allmaras model is a more recent and simpler isotropic eddy-viscosity model as it solves only one transport equation of an auxiliary variable $\tilde{\nu}$:

$$\frac{\partial \tilde{\nu}}{\partial t}+\frac{\partial u_j \tilde{\nu}}{\partial x_j}=c_{b1}\tilde{S}\tilde{\nu}-c_{w1}f_w\left[\frac{\tilde{\nu}}{d}\right]^2+\frac{1}{\sigma}\{\nabla\cdot[(\nu+\tilde{\nu})\nabla\tilde{\nu}]+c_{b2}(\nabla\tilde{\nu})^2\} \tag{33}$$

where

$$\tilde{S}=\mid S\mid+\frac{\tilde{\nu}}{\kappa^2 d^2}f_{v2}, f_{v1}=\frac{\chi^3}{\chi^3+c_{v1}^3}$$

$$f_{v2}=1-\frac{\chi}{1+\chi f_{v1}}$$

$$\nu=\tilde{\nu}f_{v1}, \chi\equiv\frac{\tilde{\nu}}{\nu}, f_w=g\left[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right]^{1/6},$$

$$g=r+c_{w2}(r^6-r), r\equiv\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}$$

The length scale d in the above equations is selected as the distance to the solid walls.

Steady RANS models are widely used in engineering flow simulations, primarily for their low cost and fast turnover. Many three–dimensional hydraulic engineering flows have been investigated with different RANS turbulence models. For example, Wu et al. (2000) investigated sediment transportation in a 180° bend open channel by combining a version of $k-\varepsilon$ model developed by Rodi (1993) with a sediment transportation model, which considers both the suspended–load and bed–load sedimentation. The obtained numerical results, including the water–depth distribution, depth–averaged velocity distribution, the developed cross–sectional bed profile and many others, are compared with experimental observations and fairly good agreement was obtained. Olsen (2003) studied the development of a self–forming meandering channel. $k-\varepsilon$ turbulence model is used to simulate the turbulence flow field. The obtained turbulent flow field is used to calculate the sedimentation procedure, which leads to the bed level change and yielding the final formation of the meandering

channel. The simulated meandering characteristics were compared with physical model carried out at Colorado State Univ. Choi and Kang (2004) investigated open–channel flows with different vegetation conditions, namely submerged and emergent plants, by using Reynolds–stress model developed by Speziale et al. (1991). These examples show how useful RANS models are for relatively simple flows. However, when these models are applied to complex flows dominated with separation and coherent vortices, results are mixed. Jones et al. (2001) compared the performance of six different two–equation turbulence models, including three versions of $k-\varepsilon$ models and three $k-\omega$ models, in predicting turbulent flows in a impeller stirred tank by comparing with experimental data. In regions far from the impeller, good predictions were obtained by these turbulence models. While in region near the impeller shaft and in the impeller discharge region, all models failed to accurately reproduce the flow. Flynn and Eisner (2004) used the standard $k-\varepsilon$ model to study the flow around an elliptical cylinder. Results showed good agreement with experiments in regions far from the objects, but poor near region simulations were obtained. Similar failure was also reported by Richmond-Bryant (2003). Such failures are not unexpected. Actually applications of steady RANS models on complex flows dominated with large–scale coherent vortices are anticipated to fail, because the majority of these RANS models are calibrated by reference to key steady flows. As shown in Eq. (27), fluid motions in these flows can be decomposed into three parts: the time–averaged component U_i, the time–periodic motion $\widehat{u_i}$ and the turbulence fluctuations u'. The time–periodic fluctuation $\widehat{u_i}$ is strongly flow and geometry dependent so does its effects on the time–averaged flow. These effects by no means can be accurately simulated by those RANS models calibrated by steady turbulent flows. Moreover, since these motions are strongly geometry dependent, models calibrated for one geometry can not be applied to other flows without modification, thus directly resolving these large–scale motions is a sounding solution to such flow simulations.

2) **Unsteady RANS**

Unsteady RANS (URANS) modeling relies on the previously discussed triple decomposition given by Eqn. (27). In flows with large-scale organized unsteadiness, the primary

benefit from URANS is that it resolves directly $\widehat{u}_i$, thus, alleviating the burden on the turbulence model, which only has to account for the random turbulent part. In a URANS model the mean and coherent flow components in Eqn. (27) are lamped together and the decomposition of instantaneous quantities reads as follows:

$$u_i\left(t,\overrightarrow{x}\right) = U_i\left(t,\overrightarrow{x}\right) + U_i'\left(t,\overrightarrow{x}\right)$$

where:

$$U_i = \overline{u_i}\left(\overrightarrow{x}\right) + \widehat{u}_i\left(t,\overrightarrow{x}\right)$$

and

$$U_i' = u_i'\left(t,\overrightarrow{x}\right)$$

Using the above decomposition the unsteady RANS (URANS) equations can be derived for the phase-averaged velocity component U_i as follows:

$$\begin{aligned} \frac{\partial U_i}{\partial x_i} &= 0 \\ \frac{\partial U_i}{\partial t} + \frac{\partial U_i U_j}{\partial x_j} &= -\frac{\partial p}{\partial x_i} + \nu\nabla^2 U_i + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left(-\rho\overline{U_i' U_j'}\right) \end{aligned} \tag{34}$$

The Reynolds stresses in the URANS equations are typically modeled using models developed for the steady RANS equations. Before proceeding with a review of recent work in the area of URANS modeling it is important to note that triple decomposition of the velocity field allows us to calculate the total turbulence kinetic energy as follows:

$$\begin{aligned} k &= \frac{u_i u_i}{2} - \frac{\overline{u_i}\,\overline{u_i}}{2} \\ &= \frac{\overline{\widehat{u}_i\ \widehat{u}_i}}{2} + \frac{\overline{u_i' u_i'}}{2} \end{aligned} \tag{35}$$

It is seen in the above equation that the essence of the tri–component decomposition is to divide the turbulence kinetic energy into two–components, that due to the large–scale, deterministic fluctuations and that due to small–scale unresolved turbulence fluctuations.

By directly resolving the large–scale, coherent fluctuations, the energy carried by those large–scale, organized motions is directly resolved. Meanwhile, by employing a statistical turbulence model, the whole spectrum of stochastic turbulence motions could be modeled with good engineering accuracy. Such an arrangement of energy split largely alleviates the burden on the turbulence model. Early studies of Durbin (1995) show that the energy carried on the geometry–induced large–scale motions is in the same order as those by turbulence fluctuations.

Earlier URANS simulations focused on 2D flows (Bosch and Rodi, 1998; Tucker and Pan, 2001; Nakayama and Miyashita, 2001; Yao et al., 2002). Since all complex flows are essentially three–dimensional, such 2D approaches are certainly not ideal. Due to the increasingly availability of powerful computational resources, 3D unsteady RANS turbulence modeling approach is gaining attention of many researchers and few 3D URANS applications have been reported in the literature. Iaccarino and Durbin (2000) and Iaccarino et al. (2003) studied 3D unsteady turbulent flows around a submerged square cylinder by using $v^2 - f$ turbulence model. The convective terms of governing equations are discretized with second–order upwinding scheme. SIMPLE algorithm is used to solve the time–dependent governing equations. A steady solution was first obtained. Later, a small asymmetric perturbation in the velocity field is imposed to excite the unsteady mode and cohere vortex shedding is obtained after several time cycles. The unsteady simulation effectively reproduced the massively separated flow physics and outperforms steady turbulence model results in the sense of flow velocity and wake prediction. In an attempt to improve the performance of unsteady RANS simulation with $k - \varepsilon$ like turbulence model, Kimura and Hosoda (2003) developed a non–linear realizable $k-\varepsilon$ model, where the realizability constraints are derived in three types of basic 2D flows, namely, a shear flow, flow around a saddle and a focal point. Flows around a 2D square cylinder and around a 3D wall–mounted square cylinder were investigated. Improved accuracy comparing with URANS using standard $k - \varepsilon$ model was obtained. Note that in all these examples, only relatively simple geometries (square cylinder) are considered. Because of the amount of challenges facing in 3D URANS simulation in complex geometries, there are still a lot of work to do before such sophisticated and cost

demanding turbulence modeling approach can be applied to engineering flow studies.

1.2.3.2 Large–Eddy Simulation (LES)

Large–Eddy Simulation is an intermediate turbulence modeling approach between RANS and DNS. The basic idea of LES is broadly similar to that of the RANS approach. The instantaneous flow field is also decomposed into a resolved (filtered) part that can be directly resolved with reasonable computational cost and an unresolved part, whose statistical influence on the resolved motions need to be accounted for using a model. However, unlike the time–averaging filter used in the RANS approaches, LES employs spatial filtering to define the resolved field as follows:

$$\widetilde{U}_i = \int_D G(x, x', \Delta) u_i(x', t) dx' \tag{36}$$

where G is the filtering operator, Δ is the filter width typically taken to be proportional to the computational grid spacing, and $\widetilde{U}_i$ indicates the filtered flow field variables. Such a filtering procedure leads to the following decomposition of the flow variables:

$$u = \widetilde{U}_i + u' \tag{37}$$

where u' is referred to as the *SGS (sub-grid scale) motions* or *residual motions* and represents the part of fluid motion that is smoothed out by the spatial filter.

The filtered equations used in LES are obtained by using the above decomposition and applying the spatial filter (Eq. 36) to the Navier-Stokes (Eq. 1) and can be written as follows

$$\frac{\partial \widetilde{U}_i}{\partial x_i} = 0 \tag{38}$$

$$\frac{\partial \widetilde{U}_i}{\partial t} + \frac{\partial \widetilde{U}_i \widetilde{U}_j}{\partial x_j} = -\frac{\partial \widetilde{p}}{\partial x_i} + \nu \nabla^2 \widetilde{U}_i - \frac{1}{\rho} \frac{\partial \widetilde{T_{ij}}}{\partial x_i} \tag{39}$$

where

$$T_{ij} = \rho \left(\widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j \right) \tag{40}$$

is the subgrid–scale (SGS) stress tensor, responsible of momentum exchange between the subgrid- and resolved scales of motion.

By construction LES will resolve directly all turbulent scales that can be represented by a given mesh resolution and account for sub-grid scale motions by an appropriate *SGS* model. Assuming that the computational mesh is sufficiently fine, LES could resolve directly most of the energy containing scales, which are the ones that are strongly dependent on the geometry of the flowfield. The SGS scales, on the other hand, if sufficiently small will tend to be more universal and thus could be modeled with a relatively simple SGS closure model. Several different SGS models have been developed but by far the most popular one is the Smagorinsky model, in which the SGS stress is expressed as

$$T_{ij} = \frac{1}{3} T_{kk} \delta_{ij} - 2\nu_T \tilde{S}_{ij} \tag{41}$$

where $\tilde{S}_{ij}$ is the resolved strain–rate tensor,

$$\tilde{S}_{ij} = \frac{1}{2} \left(\frac{\partial \tilde{U}_i}{\partial x_j} + \frac{\partial \tilde{U}_j}{\partial x_i} \right)$$

and the scalar eddy viscosity ν_t is given by

$$\nu_T = (C_s \Delta)^2 \mid \tilde{S} \mid$$

where C_s is the Smagorinsky coefficient and

$$\mid \tilde{S} \mid = \sqrt{2 \tilde{S}_{ij} \tilde{S}_{ij}}$$

A dynamic version of this model has also been proposed (Germano et al., 1991), which determines the Smagorinsky coefficient as part of the calculation using input from the smallest resolved scales. For a comprehensive recent review of various LES models the reader is referred to Piomelli (1999).

In general LES has been shown to be a very powerful modeling tool in flows dominated by large-scales, such as in free shear flows where LES on a sufficiently fine mesh can produce very accurate prediction of the turbulence flow field (Pope, 2004). Good results have also been obtained for wall bounded flows (Piomelli, 1999) but in this case the grid resolution requirements for a highly-resolved LES tend to become comparable to a DNS. Chapman (1979), for instance, estimated that the grid nodes requirement for LES simulation of plane–channel is proportional to $Re_L^{1.8}$, which is very close to the requirements of

DNS. Costs for more complex geometries problems are expected to be much higher and can't be estimated *a priori*. It is important to note, however, that even though LES may not be the modeling tool of choice for most engineering applications at realistic Reynolds numbers it holds great promise for cardiovascular flow problems, which typically occur in small to moderately high Reynolds numbers. The studies conducted by Mittal et al. (2001, 2003) clearly demonstrate what role LES can play in cardiovascular flow simulations. These studies investigate the pulsatile flows in a simplified model of stenosed artery over a range of peak Reynolds–numbers from 750 to 2000. Flows with low Reynolds numbers ($Re \leq 1400$) are studied with DNS and the rest LES. Laminar inflow condition is specified, with the velocity profile obtained from the analytical solution to Womersley flows. The LES simulations accurately predict the mean flow velocity distributions. In addition to that, the time–accurate simulations also reproduce a number of flow features observed in experiments. A series of Kelvin–Helmholtz type vortices are captured in the shear layer downstream of the constriction (modeling stenosis) for flows at Re 1000. Two shear layers were obtained downstream of the constriction. The obtained DNS and LES solutions are further analyzed with statistic analysis to illustrate the turbulence structures and spectral characteristics of these flows. However, these studies are focused on a simplified model geometry (a channel with a one–sided semicircular constriction). Extensive researches and developments are required before LES can be applied to physiology flow simulations.

1.2.3.3 Hybrid URANS/LES models

Because of the previously discussed inherent limitations of pure RANS and LES methodologies recent work is focusing on hybrid formulations, which attempt to combine the strengths of both approaches into a single, computational efficient modeling framework. As stated earlier, in regions away from the solid walls, LES modeling is well understood and can provide good descriptions of the large, energy containing scales of motion. However, when applied to wall–bounded flows, the computational cost doesn't differ significantly from that of DNS (Chapman, 1979; Spalart et al., 1997; Baggett et al., 1997). To circumvent the severe near–wall resolution requirement, one alternative is to combine LES with wall–layer

models. A relatively coarse grid is used to carry out LES simulation on wall–bounded flows. The effects of energy containing eddies in the near wall region is modeled from a wall model calculation. Promising results have been obtained through such a wall–modeling approach (Wang and Moin, 2002). However, as shown by Leschziner (2000), solutions from LES with wall modeling are sensitive to the near–wall treatment and the errors arising from the numerical approximation and the SGS model. Moreover, LES models, even with well–tuned wall models, are still too expensive for large–scale deployment in complex engineering flow simulations (Spalart, 2000).

Hybrid URANS/LES modeling strategies are essentially LES models designed to asymptote to a URANS model near solid walls. That is, in regions far from the solid walls, the well understood LES model is retained to resolve the large–scale motion while in regions near solid walls the simple, and widely tested RANS models are employed. Because of the relatively low resolution requirements of RANS models, the computational requirement in near wall regions is much lower than that of LES models, thus, largely decreasing the overall computational cost. Meanwhile, by retaining an LES simulation in the core flow region, the turbulence kinetic energy carried by large–scale motions (scale larger than the filter length scale) are resolved through the numerical simulation. Currently, there are two popular hybrid URANS/LES approaches: 1) two layer model proposed by Balaras and Benocci (1994); Balaras et al. (1996); and 2) Detached–eddy Simulation (DES) proposed by Spalart et al. (1997).

The two layer model was proposed by Balaras and Benocci (1994). In order to circumvent the extremely high grid resolution requirement of LES in the near wall region, the flow domain is divided into two different layers: one core layer and one inner–layer. In the core flow region, a coarse mesh is employed and LES is used to resolve the flow turbulence in this region. In the near wall region, simulations are carried out on a grid that is embedded on the coarse LES grid and refined in the wall–normal direction. Solutions on this inner–layer grid are obtained through URANS simulations. The TLM model uses the boundary–layer

equations in the inner–layer:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_n u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_n}\left[(\nu + \nu_t)\frac{\partial u_i}{\partial x_n}\right] \tag{42}$$

where n is the normal direction and i indicates the streamwise and spanwise directions. In Balaras and Benocci (1994), the eddy–viscosity in the above equation is provided through an algebraic eddy–viscosity model,

$$\nu_t = (\kappa y)^2 D(y) \mid S \mid \tag{43}$$

where y is the distance from the solid wall, $\mid S \mid$ is the magnitude of the resolved strain–rate tensor, and $D(y)$ is a damping function that assures the behavior of ν_t at the wall:

$$D(y) = 1 - exp[-(y^+/A^+)^3] \tag{44}$$

where $A^+ = 25$. In the TLM model, the LES and URANS simulations are combined together through communications on the layer interfaces. The URANS equations are solved with no–slip boundary condition at the solid–walls and the freestream side boundary condition is given by the LES solutions. The solid–wall effects on the core flow, on the other hand, is modeled by using the wall shear stress obtained in the inner–layer calculation as the boundary condition of LES calculation.

Another zonal approach is the Detached Eddy Simulation (DES) approach, which was proposed by Spalart et al. (1997). It is aimed primarily at massively separated flow and has recently attracted considerable attention due to its simplicity and preliminary success in various complex flows (Squires et al., 2001; Hedges et al., 2002; Paik et al., 2004). Similar as the TLM, DES combines the RANS simulation in the boundary–layers with the solution of LES in the separated regions where the detached eddies are important. However, unlike the separate mesh and different sets of equations required by the TLM approach, DES solves a single set of equations on a single grid system. Such an arrangement makes DES an attractive approach for complex engineering flow simulations, since there is no need for *a priori* division of inner–layer and out–layer, which could be very difficult if not impossible in complex geometries.

The DES method uses the same equations as Spalart–Allmaras URANS model. The only difference lies on the selection of the length scale d. In SA model, a single RANS length scale, which is the distance to solid walls, is used throughout the calculation domain. While in DES, an LES length scale is introduced and the minimum between the RANS and LES length scales is selected as the DES length scale d.

$$d = min(d_{RANS}, d_{LES}) \tag{45}$$

where $d_{LES} = C_s max(\Delta x, \Delta y, \Delta z)$ and $C_s = 0.65$.

DES is a relatively new model and is and still developing (Piomelli et al., 2003). There have been only few success applications of DES in engineering flow simulations. However, its' superiority over URANS models has been clearly demonstrated in these applications. Earlier DES applications were focused on relatively simple geometries, such as airfoils (Shur et al., 1999), cylinders (Travin et al., 2000), and spheres (Constantinescu and Squires, 2003, 2004). Results from these simple geometries are highly encouraging. In order to investigate the capability of URANS and DES methods in complex, massively separated flow simulations and to understand the natural of physical modeling obtained with these two methods, Hedges et al. (2002) investigated the turbulent flows around a simplified landing gear truck, whose geometry is based on a 31% scale Boeing 757 main landing bogie. The simulated results were compared with experimental data in the sense of pressure distribution, overall flow features, and lift and drag coefficient. Both URANS and DES yield reasonable prediction of these mean flow quantities. In order to further assess these two methods, the instantaneous solutions and time–averaged results from both methods are compared with each other. Comparisons show that time–averaged results from both methods are close to each other. In terms of the instantaneous flow field, DES solutions consistently out–perform the URANS results. DES flow–field shows a much more dynamical vortical structure than the URANS solutions, captures more large–scale unsteady kinetic energy than URANS, and reproduces a more realistic high–frequency signal distribution than URANS solutions. Applications of DES in more complex geometries can be found in Forsythe et al. (2004), where RANS and DES are used to simulate the flow around an F-15E

at high angle of attack. Both methods predict the lift and drag coefficient of the aircraft with excellent accuracy, with DES produces more favorable result than RANS solutions. In another application by Paik et al. (2004), DES is also extended to the simulation of hydraulic engineering flows. Flows past a corner mounted rectangular block and flows in a strongly curved rectangular bend are studied by means of DES and URANS. The results clearly illustrate the promise of DES method over URANS. For example, in the simulations of flows around a corner mounted rectangular block, DES results successfully reproduced all flow features observed in experimental flow visualizations. On the other hand, URANS yielded an unsteady solution to the flow field but failed to capture the rich dynamics in the vicinity of the sharp–corner object.

1.2.4 Parallel computation

Parallel computation, which simultaneously uses multiple computer resources to solve a computational problem, is considered to be the future of high performance computation. The major reason to use parallel computing is to obtain a powerful computer system that can save computational time and also solve larger problem in an inexpensive way. Traditional computer programs are serial—one instruction executed at a time on a single processor (such as most personal computers do). To improve program performance of a serial code primarily relies on the development and deployment of faster processors. Despite of the exponential growth of power of the modern computer processors as predicted by Moore (1965), this approach obviously has its limitations. The development of new and faster CPUs is extremely expensive and, more importantly, the current development of CPUs is approaching the physical limitation (Kish, 2002). The inexpensive alternative approach is to employ parallel computation—using multi–processors to execute different parts of the same program simultaneously. By combining the power of a large number of processors, such computation strategy can drastically decrease the physical time used for computation. Meanwhile, single process machines always come with a limited amount of memory, thus imposing a physical limitation on the size of the problem that can be solved. By using multiple computer resources, the memory size can be linearly increased thus increase the

size of problem that can be solved.

High fidelity computational fluid dynamics typically requires a very fine grid system to provide enough resolution to the flow domain and also needs time–dependent solution of the governing equations thus the actual flow physics are accurately reproduced numerically. Such CFD simulations are extremely cost–demanding and are at the limits of computational power. The major part of any CFD computation is residual calculation at discretized node in space and the solution of a linear matrix system which is used to iterate the solution to a converged state. These calculations are almost independent on each other thus are suitable for parallel computation. Parallel computation strategy is adopted by almost all commercial CFD software packages, including Fluent, CFX, PHOENICS, etc., and a vast number of researchers (Gropp et al., 2001; Djomehri and Biswas, 2003; Takaki et al., 2003; Dong and Karniadakis, 2004).

The idea of parallel computation seems rather simple and beautiful, however, the implementation is not. A well designed parallel program should yield the same result as that obtained from single processor computations (correctness) and should perform well with the addition of extra processors (scalability). To achieve this goal is fairly complex and error prong. To alleviate the difficulties of parallel programming, many programming libraries and toolkits, such as MPI (message passing interface), PVM (parallel virtual machine), OpenMP, and Petsc (portable, extension toolkit for scientific computation) etc., were developed in the past researches. Current parallelization task primarily depends on these libraries and toolkit.

Selection of parallel programming methods and libraries depends on the memory architecture of parallel computer system. Modern parallel computer systems can be divided into two groups: shared memory multi–processors and distributed memory multi–processors. Shared memory system is a natural extension of traditional single processor model. Multiple processors are connected to multiple memory modules such that each processor can equally access the memory modules. Only a single memory address is used and the system allows processor communication through variables stored in a shared memory address.

The later one is constructed by connecting a cluster of computers with high–speed communication networks. Since the variables are equal accessible to any processor on a shared memory system, programming on these systems is much more convenient than on a distributed memory one. The major limitation of shared memory system is when processor number reaches a certain number, the hardware performance gain from extra processor will drastically decrease thus scalability can not be maintained, no matter how good the program is coded. On the other hand, the distributed memory system, which always appears as a cluster of workstations, can gain better scalability performance by introduction of extra computation nodes even on massively parallel system (which has very large number of processors). With the appearance of powerful while inexpensive CPUs in the later 1990s, this approach is more attractive than anytime before. Almost all today's top super computers (http://www.top500.org) are using this technique. However, to program on this category of parallel system requires carefully parallel strategy design, work load distribution, load balancing and data communication. Almost the whole serial computational code need be rewritten. Despite its' complex programming procedure, clusters are increasingly adopted (http://www.top500.org) because of their relative low cost and commercial CFD codes are parallelized on these systems.

The computational facilities available for this research are several SGI Origin systems with multi–processors. These systems are shared memory computers. Therefore, the following review will focus on shared memory parallel programming. Although there are many available programming methods for these systems, the practical approach is to use OpenMP, an industry standard supported by all major computer manufactures (http://www.openmp.org). OpenMP is a collection of compiler directive (which can be inserted into the source code to tell the compiler that the following calculation should be executed in a parallel manner), library routines, and environmental variables. All these compiler directive and library routines can be directly inserted into the original sequential program which will yield a parallel version of the original code. The execution model of OpenMP programs is shown in Fig. 5. It starts like any traditional sequential program with only one thread. In regions where parallel computations are allowed (for example, most DO

loops), user can start a parallel region to create additional threads to execute the program simultaneously. The additional threads will go away when the region is finished and can be started later when it is needed. This is also called *fork–join model*. The task of programming is focused on the design of parallel region and sharing of the data thus greatly alleviates the programming burden. It is also worth to note that the users of OpenMP approach rely on compiler to perform the actual parallelization job (to divide the load, communicate between processors, etc.). Users can directly benefit from the state–of–the–art computer science research results by using the most up-to-date compilers, instead of the learning–&–implementing approach, which requires much more computer science knowledge. Because of these benefits, OpenMP are widely used in scientific computations, for example, in molecular dynamics (Zhou et al., 2003), numerical algorithm development (Takahashi et al., 2003), air pollution investigation (Martin et al., 2004), and also CFD (Jin et al., 2000).

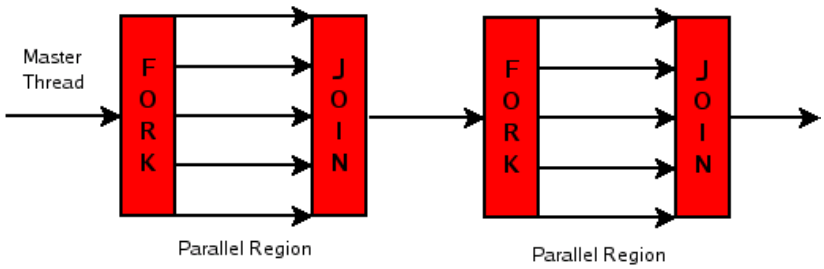


Figure 5: OpenMP's fork–join parallel model

1.3 Objectives

The objective of this thesis is to develop, validate, and demonstrate the versatility of a powerful computational methodology for simulating complex, three-dimensional engineering flows using unsteady statistical turbulence models. The developed methodology is intended to:

1) be a reliable and efficient engineering tool that can be applied to carry out unsteady simulations of flows in complex 3D geometries discretized with fine computational meshes (millions of grid nodes);

2) be capable of simulating flows in domains with geometrical features at disparate

spatial scales;

3) be applicable to a broad range of engineering problems, from hydraulics to biofluids; and

4) provide the computational framework in which scalar and/or particulate transport, multi-physics models can be incorporated in the future in order to study, say, sediment transport processes in natural rivers or blood cell transport in cardiovascular flows.

To demonstrate the wide applicability of the method, we apply it to simulate and analyze the physics of two widely different flow problems, one from hydraulics and the other from cardiovascular fluid mechanics.

The first problem is flow around bridge foundations. This study is primarily stimulated from the needs of bridge scour prediction. The introduction of bridge foundations, such as piers and abutments, in a natural river reach alters drastically the local hydraulics inducing large–scale unsteadiness and shedding of coherent vortices. Whirlpool, tornado, and necklace (or horseshoe) vortices have all been detected in laboratory studies of bridge foundation flows and have been implicated in the onset of sediment transport, bed erosion and scour (Melville, 1997; Chrisohoides et al., 2003). Since continuous scouring of the streambed could cause serious damage on bridge support and even cause the bridge to fail, the problem of bridge scour has been and continues to be at the center of ongoing research by hydraulic engineers. To date, no numerical method exists capable of quantitatively accurate predictions of the complex, unsteady coherent vortices induced by real-life bridge foundations at realistic Reynolds numbers.

The second problem is flow in mechanical, bileaflet heart valves. Approximately 170,000 individuals worldwide receive prosthetic heart valves every year, and over half (55%-65%) receive mechanical heart valves (MHVs). Fig. 6 shows a picture of St. Jude Medical mechanical heart valve. Recipients of MHVs, however, must take anticoagulant medication because of the potential for thromboembolic complications. Such complications are thought to be caused by high blood shear stresses, turbulence, and the overall complexity of the hemodynamics in MHVs. Same as the bridge foundation problem, no numerical methods are available today capable of simulating the complex, unsteady flow patterns in MHV.

Figure 6: St. Jude Medical mechanical heart valve.

Even though seemingly disparate, both flow problems are characterized by complex geometries. Typical bridge foundation flows occur in multi–connected, arbitrarily shaped domains, while as shown in Fig. 6, the flow in aorta is blocked with the bileaflets structure. Both flows can be categorized as flow around bluff bodies. Such flows are dominated by vortex shedding, and large–scale unsteadiness. These complexities cast big challenges in numerical simulation of such flows. In this thesis, efforts are focused on addressing two important issues in answering these challenges: 1) the ability to resolve the complex geometry with reasonable efforts; and 2) the ability to accurate reproduce the complex unsteady flow dynamics. The following outlines the specific tasks of this thesis:

1.3.1 Development of an accurate and efficient numerical method for URANS simulation in complex geometries

The numerical method in this work is based on the overset, Chimera domain decomposition method of Tang (2001). The governing equations are formulated in generalized curvilinear coordinates, discretized with second-order accurate finite-volume schemes, and integrated in time using a dual-time artificial compressibility approach. It is important to note that in Tang (2001) the governing equations were integrated in pseudo–time with an

explicit Runge–Kutta scheme. In order to obtain smooth solutions across interfaces, a new interpolation scheme based on enforcing a second-order accurate discrete approximation of global mass conservation at each overset grid interface was developed in Tang (2001). The numerical method was applied to many laminar flows and good results were obtained. However, the emphasis of this work is on turbulent flows using unsteady statistical turbulence models. Extensive numerical tests showed that when applying the method of Tang (2001) to solve the URANS and turbulence closure equations, numerical instabilities develop at grid interfaces during the early stages of the dual time-stepping iteration scheme. These instabilities were linked to the production terms in the turbulence closure equations, which depend on products of velocity gradients. Such terms impose more stringent than in laminar flow simulations smoothness requirements for both the velocity field and its gradients across grid interfaces. In fact we found that in order to avoid numerical instabilities during the early stages of the simulation, a smooth velocity field across interfaces needs to be established within as few dual time stepping iterations as possible. In this work, a diagonalized, block-implicit, approximate–factorization scheme is developed and applied to integrate the discretized governing equations in pseudo–time implicitly. Various turbulence modeling options are incorporated into the flow solver, including URANS with the standard $k-\varepsilon$ model, URANS with the Spalart-Allmaras model, and DES. For all turbulence models employed the implicit iterative scheme developed in this thesis is shown to yield smooth and stable unsteady solutions in very complex computational domains. The computer code is also parallelized using OpenMP and its efficiency and scalability is demonstrated.

1.3.2 Numerical sensitivity studies and model validation

Extensive numerical tests are carried out to investigate the sensitivity of the computed solutions to various numerical parameters, such as the size of the computational grid, the time step, and the extent of the computational domain. Experimental measurements for both the bridge foundation flow and the MHV problem are employed to validate the numerical model. These comparisons not only establish the predictive capabilities of the model but also help improve the laboratory experiments by underscoring potential sources

of experimental uncertainty.

1.3.3 Elucidation of flow physics

The simulated solutions are analyzed to elucidate the complexity of flow patterns in the two problems under consideration. For the bridge foundation flow the emphasis is placed on establishing links between the complex hydrodynamics and the observed in laboratory experiments scour patterns in the vicinity of an identical bridge foundations. For the heart valve flow we examine the evolution of the flow with Reynolds number and discuss the potential significance of the observed flow phenomena from the stand point of the transport of blood elements.

The thesis is organized as follows: in Chapter 2 we present the developed numerical method. The discretization and time–advacing scheme of governing equations and turbulence equations will be covered. Implementation of overset grid method and parallelization with OpenMP will also be discussed. The next two chapters are devoted to our numerical studies on the selected two engineering flow problems, with each chapter covers one problem. Finally we present conclusions based on our numerical results.

CHAPTER II

NUMERICAL METHODS

In this chapter, we present the numerical method we have developed for simulating complex turbulent flows. The chapter is organized as follows. First we present the governing equations in generalized curvilinear coordinates, including both the mean flow and turbulence closure equations. Next we describe the base solve for integrating numerically the time-accurate governing equations. Subsequently we present the overset grid domain decomposition method for simulating arbitrarily complex geometries. Finally we discuss some issues related to the OpenMP compiler directive, which we used to develop a parallelized version of the computational code.

2.1 Governing equations

We solve numerically the unsteady Reynolds–averaged Navier–Stokes equations in conjunction with various turbulence modeling options. The URANS equations are transformed in generalized, non-orthogonal curvilinear coordinates and formulated in strong conservation form as follows:

$$\Gamma \frac{\partial Q}{\partial t} + J \frac{\partial}{\partial \xi^j} \left(F^j - F_v^j \right) = 0 \tag{46}$$

where

$$\Gamma = \mathrm{diag}\,[0\,1\,1\,1]$$

$$Q = [p, u_1, u_2, u_3]^T$$

$$F^j = \frac{1}{J} \left[U^j, u_1 U^j + p\xi^j_{x_1}, u_2 U^j + p\xi^j_{x_2}, u_3 U^j + p\xi^j_{x_3} \right]^T$$

$$F_v^j = \frac{1}{J} \left(\frac{1}{Re} + \nu_t \right) \left[0, g^{mj} \frac{\partial u_1}{\partial \xi^m} + R_{m1}\xi^j_{x_m}, g^{mj} \frac{\partial u_2}{\partial \xi^m} + R_{m2}\xi^j_{x_m}, g^{mj} \frac{\partial u_3}{\partial \xi^m} + R_{m3}\xi^j_{x_m} \right]^T$$

In the equations above, p is the static pressure divided by the density, u_i $(i = 1, 2, 3)$ are the Cartesian velocity components, x_i are the Cartesian coordinates, J is the Jacobian of the geometric transformation, $\xi^j_{x_i}$ are the metrics of the geometric transformation, U^j are the

contravariant velocity components $U^j = u_i \xi^i_{x_j}$, g^{ij} are the components of the contravariant metric tensor $g^{ij} = \xi^i_{x_k} \xi^j_{x_k}$, the tensor R_{ij} is defined as

$$R_{ij} = \frac{\partial u_i}{\partial \xi^k} \xi^k_{x_j}$$

and $\Re$ is the Reynolds number. Note that the Boussinesq approximation has been used in the above equations to express the Reynolds stresses in terms of the mean rate of strain via the turbulent eddy viscosity ν_t. In this work, three different turbulent models are employed to account for turbulence effects. A detail description of the turbulence models and their discretization will be discussed in the next section.

2.1.0.1 Spatial and temporal discretization

The URANS equations [Eq. (46)] are discretized with a second-order-accurate finite-volume scheme. The time derivative is discretized with second-order accurate, three-point-backward, Euler-implicit temporal-integration scheme while the spatial derivatives are discretized in strong-conversation form using central differencing with explicitly added artificial dissipation terms. The resulting discrete approximation of [Eq. (46)] reads as

$$\Gamma \frac{1}{J} \frac{3Q^{n+1}_{i,j,k} - 4Q^n_{i,j,k} + Q^{n-1}_{i,j,k}}{2\Delta t} + \left(\delta_{\xi^m} \widetilde{F}^m - \delta_{\xi^m} \widetilde{F}^m_v \right)^{n+1}_{i,j,k} \;=\; 0 \tag{47}$$

where

$$\delta_{\xi^1} (\,)_{i,j,k} = \frac{(\,)_{i+1/2,j,k} - (\,)_{i-1/2,j,k}}{\Delta \xi^1} \,, \tag{48}$$

the superscript n indicates the time level and Δt is the time step. The flux $\widetilde{F}$ denotes the numerical approximation of the flux F at the cell interfaces, which is evaluated as follows:

$$\widetilde{F}^1_{i+1/2,j,k} = \frac{1}{2} \left(F^1_{i+1,j,k} + F^1_{i,j,k} \right) \;+\; Diss^1_{i+1/2,j,k} \tag{49}$$

where $Diss^1$ is an artificial dissipation flux. Lin and Sotiropoulos (1997a) evaluated several approaches for constructing dissipation fluxes including scalar- and matrix-valued artificial dissipation and flux-difference-splitting upwinding. They showed that when all schemes are compared on the same mesh resolution the matrix-valued scheme (Lin and Sotiropoulos, 1997a,b) introduces the least overall amount of artificial dissipation. In this work, we use

the matrix-valued scheme and calculate the dissipation flux as follows:

$$Diss^1_{i+1/2,j,k} = \eta\delta_{\xi^1}\left(\left|A^1\right|\delta_{\xi^1}\delta_{\xi^1}\right)Q_{i+1/2,j,k} \tag{50}$$

where $\left|A^1\right|$ is the absolute value of the Jacobian matrix $A^1 = \partial F^1/\partial Q$ and η is a constant that controls the amount of artificial dissipation. The absolute value of the Jacobian matrix is defined as follows:

$$\left|A^1\right| = M_j^{-1}\left|\Lambda^1\right|M_j \tag{51}$$

where M is the modal matrix of A^1 and $\left|\Lambda^1\right|$ is the diagonal matrix with entries the absolute values of the eigenvalues of A^1. For incompressible URANS equations the Jacobian matrices A^j ($j = 1$ to 3) are given as follows:

$$A^j = \frac{1}{J}\begin{bmatrix} 0 & \beta\xi^j_{x_1} & \beta\xi^j_{x_2} & \beta\xi^j_{x_1} \\ \xi^j_{x_1} & U^j + u_1\xi^j_{x_1} & u_1\xi^j_{x_2} & u_1\xi^j_{x_3} \\ \xi^j_{x_2} & u_2\xi^j_{x_1} & U^j + u_2\xi^j_{x_2} & u_2\xi^j_{x_3} \\ \xi^j_{x_3} & u_3\xi^j_{x_1} & u_3\xi^j_{x_2} & U^j + u_3\xi^j_{x_3} \end{bmatrix}$$

The eigenvalues of the Jacobian matrices are as:

$$\lambda^j_1 = U^j - C_j, \quad \lambda^j_2 = U^j + C_j, \quad \lambda^j_3 = \lambda^j_4 = U^j$$

where $C_j = \sqrt{(U^j)^2 + \beta g^{jj}}$. The modal matrix, its inverse and $|A^j|$ are given in Appendix A.

Finally, the calculation of the viscous fluxes at cell interfaces is straightforward. The metrics of the geometric transformation were calculated using simple averaging, and the spatial derivatives of the velocity field were calculated using three-point, central differencing (see Lin and Sotiropoulos (1997a) for details).

2.1.1 Dual-time stepping artificial compressibility algorithm

Since Γ in Eq. 47 is a singular matrix, the discrete equations cannot be directly advanced in time to the next time step. Moreover, the temporal-discretization scheme is fully implicitly

so advancing Eq.47 in time requires an iterative algorithm. Such an algorithm can be designed by adopting the dual- (or pseudo-) time-stepping, artificial compressibility approach (Merkle and Athavale (1987); Tang et al. (2003)). In this section we describe the dual-time iterative algorithm we have developed for solving the URANS. It's worthwhile to note that similar scheme is also applied to solve the turbulence equations. A pseudo- or dual-time derivative term is added to the discrete governing equations and the resulting equations can be written as follows:

$$\frac{1}{J}\frac{Q^{l+1}_{i,j,k}-Q^{l}_{i,j,k}}{\Delta\tau}+\Gamma\frac{1}{J}\frac{3Q^{l+1}_{i,j,k}-4Q^{n}_{i,j,k}+Q^{n-1}_{i,j,k}}{2\Delta t}+\left(\delta_{\xi^m}\widetilde{F}^m-\delta_{\xi^m}\widetilde{F}^m_v\right)^{l+1}_{i,j,k} \;=\; 0 \qquad (52)$$

where τ is the pseudo time variable, and l denotes the iteration level in pseudo-time. The above equations need to be iterated in pseudo-time until the pseudo-time derivatives are reduced to a small tolerance, Q^l and $Q^{l+1}\rightarrow Q^{n+1}$, and the original governing equations are satisfied at the new time level $n+1$. In this work we adopt the block-implicit, approximate factorization method of Beam and Warming (1976), which is adapted to unsteady flow as follows.

The discrete dual-time equations can be reorganized as follows:

$$\frac{1}{J}\left(\frac{\Delta Q_{i,j,k}}{\Delta\tau}+\Gamma\frac{3}{2}\frac{\Delta Q_{i,j,k}}{\delta t}\right) \;=\; -R\left(Q^{l+1}_{i,j,k}\right) \qquad (53)$$

where $\Delta Q_{i,j,k}=Q^{l+1}_{i,j,k}-Q^{l}_{i,j,k}$

$$R\left(Q^{l+1}_{i,j,k}\right) \;=\; \Gamma\frac{1}{J}\frac{3Q^{l}_{i,j,k}-4Q^{n}_{i,j,k}+Q^{n-1}_{i,j,k}}{2\Delta t}+\left(\delta_{\xi^m}\widetilde{F}^m-\delta_{\xi^m}\widetilde{F}^m_v\right)^{l+1}_{i,j,k}$$

The term $R\left(Q^{l+1}\right)$ is linearized in pseudo-time as

$$R\left(Q^{l+1}_{i,j,k}\right) \;=\; R\left(Q^{l}_{i,j,k}\right)+\mathfrak{L}^l\Delta Q_{i,j,k}+O\left(\Delta\tau^2\right) \qquad (54)$$

where $\mathfrak{L}^l(\centerdot)$ is the Jacobian of the spatial residual R, $\mathfrak{L}\equiv\partial R/\partial Q$ (see below for the approach we adopt to evaluate this Jacobian). Substituting [Eq.(54)] into [Eq.(53)] and neglecting higher-order terms we obtain:

$$\frac{1}{J}\left[\left(\frac{1}{\Delta\tau}+\Gamma\frac{3}{2}\frac{1}{\Delta t}\right)I+\mathfrak{L}^l\right]\Delta Q \;=\; -R\left(Q^l\right) \qquad (55)$$

The resulting equation system is a large sparse linear system that requires significant computational resources to directly invert the matrix multiplying ΔQ. To reduce the computational resources requirement, the left hand side is approximated by the following approximate factorization:

$$\left[D^l + \mathfrak{L}_3^l\right]\left[D^l\right]^{-1}\left[D^l + \mathfrak{L}_2^l\right]\left[D^l\right]^{-1}\left[D^l + \mathfrak{L}_1^l\right]\Delta Q^l \;=\; -R\left(Q^l\right) \tag{56}$$

where

$$\begin{aligned} D^l &= \left(\frac{1}{\Delta\tau} + \Gamma\frac{3}{2}\frac{1}{\Delta t}\right) I \\ \mathfrak{L}_m^l &= \delta_{\xi^m}\left(\frac{\partial \widetilde{F}^m}{\partial Q} + \frac{\partial \widetilde{F}_v^m}{\partial Q}\right) \end{aligned}$$

[Eq.(56)] is solved in the following three steps:

$$\begin{aligned} \left[D^l + \mathfrak{L}_1^l\right]\Delta Q^* &= -R\left(Q^l\right) \\ \left[D^l + \mathfrak{L}_2^l\right]\Delta Q^{**} &= D^l \Delta Q^* \\ \left[D^l + \mathfrak{L}_3^l\right]\Delta Q &= D^l \Delta Q^{**} \\ Q^{l+1} &= Q^l + \Delta Q \end{aligned}$$

In the equation above, the coefficient matrices are banded sparse, where the non-zero elements are determined by $\mathfrak{L}_m$, the derivative of the discretized convective flux $\widetilde{F}$ and viscous dissipation $\widetilde{F_v}$. In this study, the convective flux is the summation of the convective flux and fourth-order matrix dissipation. The derivative of fourth-order matrix dissipation will lead to a five–point stencil and result in a block five-diagonal coefficient matrix, which would be too expensive to invert. Since the integration over pseudo-time is continued until $\Delta Q \longrightarrow 0$, it is not necessary to use the exact derivative on the left hand side. The artificial dissipation term is replaced by a second-order derivative term scaled with the spectral radius $\rho(A)$ of A. Only the orthogonal mesh terms are retained for the implicit viscous terms in order to simplify the left-hand side of the equations. The final form of $\mathfrak{L}_m^l$ is

$$\mathfrak{L}_1^l \;=\; \frac{1}{2}\left(A^1_{i+1,j,k} - A^1_{i-1,j,k}\right)\frac{\partial}{\partial\xi^1} - \frac{1/Re + \nu_t}{J} g^{11}\Gamma\frac{\partial}{\partial\xi^1} - \eta\rho\left(A\right) I\frac{\partial}{\partial\xi^1}$$

Once again, it is important to re-iterate that none of the simplifications of the left hand side of Eq. 56 impact the converged solution as the left hand side is driven to zero at convergence during each physical time step.

2.2 Turbulence model equations

In this thesis, we employ three different turbulent models to calculate the eddy viscosity ν_t and close the URANS equations [Eq. 46]:

- Standard $k-\varepsilon$ model
- Spalart–Allmaras model
- Detached–eddy simulations

The later two models essentially share the same set of governing equations and only differ in the way of the distance to the wall is defined. Therefore, only one set of turbulent equations for these two models will be presented.

2.2.1 Standard $k-\varepsilon$ model

In the standard $k-\varepsilon$ model, the eddy viscosity is calculated in terms of the turbulence kinetic energy k and its rate of dissipation ε (Jones and Launder, 1972). The transport equations for k and ε are formulated in curvilinear coordinates as follows:

$$\frac{\partial Q_t}{\partial t} + J\frac{\partial}{\partial \xi^j}\left(F_t^j - F_{V_t}^j\right) + JH_t = 0 \tag{57}$$

where

$$\begin{aligned}
Q_t &= [k,\varepsilon]^T \\
F_t^j &= \frac{U^j}{J}[k,\varepsilon]^T \\
F_{V_t}^j &= \frac{1}{J}\left[\left(\frac{\nu_t}{\sigma_k}+\frac{1}{Re}\right)g^{jj}\frac{\partial k}{\partial \xi^j}, \left(\frac{\nu_t}{\sigma_\varepsilon}+\frac{1}{Re}\right)g^{jj}\frac{\partial \varepsilon}{\partial \xi^j}\right]^T \\
H_t &= \frac{1}{J}\left[-P_d+\varepsilon, -C_{\varepsilon 1}\frac{\varepsilon}{k}P_d + C_{\varepsilon 2}\frac{\varepsilon^2}{k}\right]
\end{aligned}$$

The production of turbulence kinetic energy P_d and the eddy viscosity ν_t are given by

$$P_d = \frac{1}{2}\nu_t \left(R_{ij} + R_{ji}\right)^2$$
$$\nu_t = C_\mu \frac{k^2}{\varepsilon}$$

The model constants assume their "standard" values: $C_\mu = 0.09$, $c_{\epsilon 1} = 1.44$, $c_{\epsilon 2} = 1.92$, $\sigma_k = 1.0$ and $\sigma_\epsilon = 1.3$.

The turbulence transport equations are discretized in space and time in a similar fashion as the URANS equations:

$$\frac{1}{J}\frac{3Q^{n+1}_{t_{i,j,k}} - 4Q^{n}_{t_{i,j,k}} + Q^{n-1}_{t_{i,j,k}}}{2\Delta t} + \left(\delta_{\xi^j}\widetilde{F}_t^{\,j} - \delta_{\xi^j}\widetilde{F}^j_{v_t}\right)^{n+1}_{i,j,k} + \widetilde{H}_t \;=\; 0 \tag{58}$$

For robustness, the convective terms are discretized using first–order upwinding differencing. The upwinding scheme is constructed by adding an artificial dissipation term to central difference discretization given as follows:

$$\left(\widetilde{F}^1_t\right)_{i+1/2,j,k} = \frac{1}{2}\left(F^1_{t_{i+1,j,k}} + F^1_{t_{i,j,k}}\right) + (Diss_t)_{i+1/2,j,k} \tag{59}$$

The artificial dissipation term is given by

$$(Diss_t)_{i+1/2,j,k} = -\frac{1}{2}(|U^j|\delta Q_t)_{i+1/2,j,k}$$

The viscous fluxes are discretized with central differencing. Finally, $\widetilde{H}_t$ in Eqn. (58) is the discrete approximation of the source term H_t computed using central differencing to discretize the velocity gradients terms in Eq. (57).

To solve the discretized $k-\varepsilon$ equations simultaneously with the URANS governing equations, we employ the same pseudo–time iteration algorithm. A pseudo–time derivative term $\partial Q_t/\partial\Delta\tau$ is added to the left hand side of the equations yielding the following set of discrete equations:

$$\frac{1}{J}\frac{Q^{l+1}_{t_{i,j,k}} - Q^{l}_{t_{i,j,k}}}{\Delta\tau} + \frac{1}{J}\frac{3Q^{l+1}_{t_{i,j,k}} - 4Q^{n}_{t_{i,j,k}} + Q^{n-1}_{t_{i,j,k}}}{2\Delta t} = RHS_t(Q^{l+1}_{t_{i,j,k}}) \tag{60}$$

where

$$RHS_t(Q^{l+1}_{t_{i,j,k}}) = -\left[\left(\delta_{\xi^m}\widetilde{F}_t^{\,m} - \delta_{\xi^m}\widetilde{F}^m_{v_t}\right)^{l+1}_{i,j,k} + \widetilde{H}_t\right]$$

Upon linearization in time, the final equations used for the pseudo–time iteration can be written as follows:

$$\frac{1}{J}\left[(\frac{1}{\Delta\tau}+\frac{3}{2}\frac{1}{\Delta t})+\mathfrak{L}_{\mathfrak{t}}\right]\Delta Q_t = RHS_t(Q_t^l)$$

where $\mathfrak{L}_{\mathfrak{t}}$ is the Jacobian matrix resulted from the above linearization. The discrete operator in the left hand side of this equation is factorized using the Alternation Direction Implicit (ADI) method and the k and ε equations are advanced in pseudo-time sequentially at the end of each pseudo-iteration for the URANS equations.

The overall solution algorithm for advancing all flow variable to the next physical time step proceeds as follows. Starting from an initial guess, the URANS equations are advanced for one pseudo-iteration by solving Eqn. (47) using the currently available eddy-viscosity field. After the new velocity field has been obtained, first the k and then the ε equations are advanced in pseudo-time by solving Eqn. 58 with ADI. The updated values for k and ε are used to calculate a new estimate for the eddy viscosity field and the entire iteration scheme starts again to advance the equations to the next pseudo-iteration level. The governing equations are so iterated in pseudo-time until convergence is reached and the solution at the next physical time step has been obtained.

2.2.1.1 Boundary conditions

In this study, the standard $k-\varepsilon$ model is used to investigate the flows around bridge piers. These piers are mounted on a rough wall, thus the bed roughness has to be considered. In this research, we use wall functions approach for this purpose as it provides a very practical alternative for calculating complex engineering flows at real-life Reynolds numbers. When generating the computational grid, the first grid surface off of a solid wall is placed within the logarithmic layer using a value for the shear velocity obtained from flat-plate, turbulent boundary-layer correlations. Because of the complexity of the flow, however, and the presence of large regions of reversed flow the first grid node off the wall often ends up within the laminar sublayer or the buffer layer—this situation is typically encountered in the

pier wakes. For that reason we employ the following, multi-layer wall-functions approach:

$$u^+ = \begin{cases} \frac{1}{\kappa}\left[\ln\left(y^+\right)\right] + B - \Delta B & y^+ > 11 \\ y^+ & y^+ \leq 11 \end{cases} \tag{61}$$

with

$$u^+ = \frac{u}{u_\tau} y^+ = \frac{u_\tau y}{\nu}$$

in which u is the magnitude of velocity at the nodes near the solid wall, u_τ is the shear velocity and y^+ is the dimensionless normal distance to the wall. The velocity shift ΔB for uniform sand-grain roughness is a function of the bed roughness height $k_s^+ = u_\tau k_s/\nu$. In this study, we use the velocity shift formulation of Cebeci and Bradshaw (1977) which reads as

$$\Delta B = \left[B - 8.5 + \frac{1}{\kappa} ln\left(k_s^+\right)\right] sin\left[0.4258\left(ln\left(k_s^+\right) - 0.811\right)\right]$$

The wall functions are implemented using the two-point approach described in detail in Sinha et al. (1998).

2.2.2 Spalart-Allmaras model and DES

As stated earlier, these two methods shared the same transport equations. The transport equation of the auxiliary variable 33 is transformed into general curvilinear coordinate system as follows:

$$\frac{\partial \tilde{\nu}}{\partial t} + \frac{\partial U_j \tilde{\nu}}{\partial \xi_j} = RHS(\tilde{\nu}) \tag{62}$$

where

$$RHS(\tilde{\nu}) = c_{b1}\tilde{S}\tilde{\nu} - c_{w1} f_w \left[\frac{\tilde{\nu}}{d}\right]^2 + \frac{1}{\sigma}\{\nabla \cdot [(\nu + \tilde{\nu})\nabla\tilde{\nu}] + c_{b2}(\nabla\tilde{\nu})^2\}$$

The constants and the definitions of the other variables can be found in Chapter 1. The above equation is solved in the similar pseudo–time manner as the k and ε equations. The convective term is discretized with the first–order upwinding scheme described in Eq. 59. The right hand side terms are all discretized with second–order central difference. In advancing the above equations in pseudo–time steps, only the convective terms are treated implicitly.

2.3 Overlapping domain decomposition method

In this section we describe the overset grid formulation we have developed for handling arbitrarily complex, multi-connected domains. To facilitate our discussion, we will present the method in the context of a problem we simulate in this work, the flow in a reach of the Chattahoochee River containing a section of the actual bridge foundation that spans the River near Cornelia, GA. The geometrical complexity of the flow domain is illustrated in Fig. 7, which shows the pier bent mounted on the bed of the Chattahoochee River. The bent consists of four rectangular piers located one behind the other along the flow direction. The geometry is further complicated by the rectangular concrete slab connecting the two middle piers, which, as shown in Fig. 7, does not extend all the way to the channel bed. During flooding events the slab is submerged and, thus, in general it needs to be taken into account. The entire bridge foundation consists of three such bents across the bridge span but here we focus on a single bent.

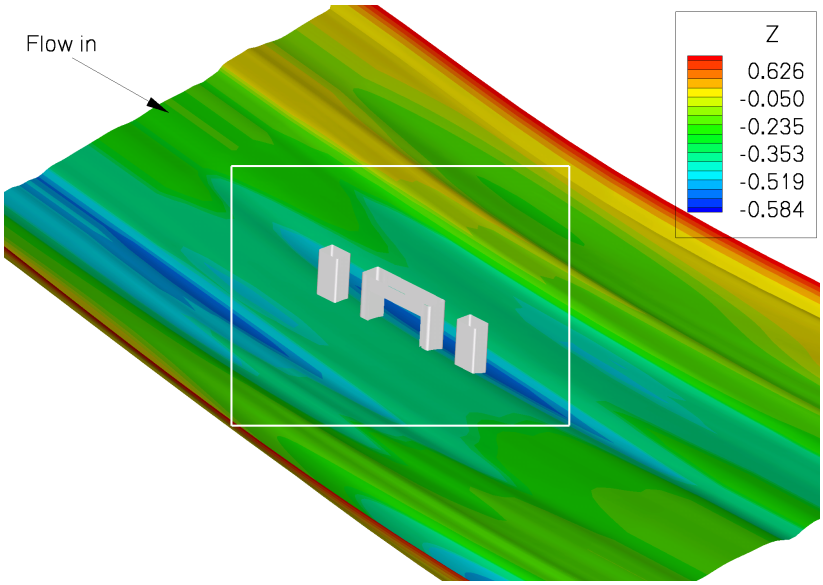


Figure 7: Bridge piers on actual river geometry

The complexity of the reach topography and that of the pier bent configuration illustrate clearly the need for the overset grid approach and the grid embedding strategy we develop in this work. The transverse and longitudinal variations of the reach bathymetry determine

the approach flow conditions and cause significant skewness of the flow relative to the pier bent. From the standpoint of predicting sediment transport and scour, on the other hand, a locally refined grid is required in the vicinity of the piers to accurately resolve the shedding of scour-inducing vortices. Grid embedding with Chimera overset grids allows us to account both for the large-scale flow features controlled by the river topography and the complex unsteady flow induced by the piers without requiring excessively large grid sizes.

As shown in Fig. 8, a curvilinear grid system, which will be referred to as the "*background grid*" hereafter, is used to discretize the river reach—denoted as *Subdomain 1* in Fig. 8. This grid is fine enough to resolve all essential features of the reach bathymetry but far too coarse to capture the foundation-induced unsteadiness. To resolve the pier-induced vortices, we employ a total of four subdomains as shown in Fig. 8. Each one of the subdomains 3, 4, and 5 is discretized with a very fine, boundary-fitted, O–type grid. Such grid layout allows us to carefully cluster grid lines near solid walls and is ideally suited for resolving the flow in the vicinity of a wall-mounted obstacle. The complex geometry of the two middle piers is accounted for by embedding a rectangular subdomain in the region between the two piers. Subdomain 6 in Fig. 8 extents vertically from the riverbed to just below the start of the slab. The grid nodes in this subdomain are optimally clustered near the two inner walls of the middle piers in the streamwise direction and near the riverbed and the slab in the vertical direction. Due to the disparity in spatial scales between the river reach and the pier bent, there is a large discontinuity between the grid spacing of the background grid (Subdomain 1) and that of Subdomains 3, 4, 5, and 6 used to discretize the piers. As shown in Tang et al. (2003) large discontinuities in mesh spacing across subdomain interfaces tends to introduce spurious oscillations in the computed flow variables and in general lead to inaccurate solutions. In order to remedy this problem, we embed yet another subdomain, Subdomain 2 in Fig. 8, between the background grid and the pier bent grids. Grid nodes are distributed uniformly in this subdomain and their total number is selected such that the spatial resolution of this grid alleviates the large discontinuity between the coarse resolution of the background grid to the very fine resolution of the pier bent grids.

The governing equations, URANS and turbulence closure equations, are advanced in

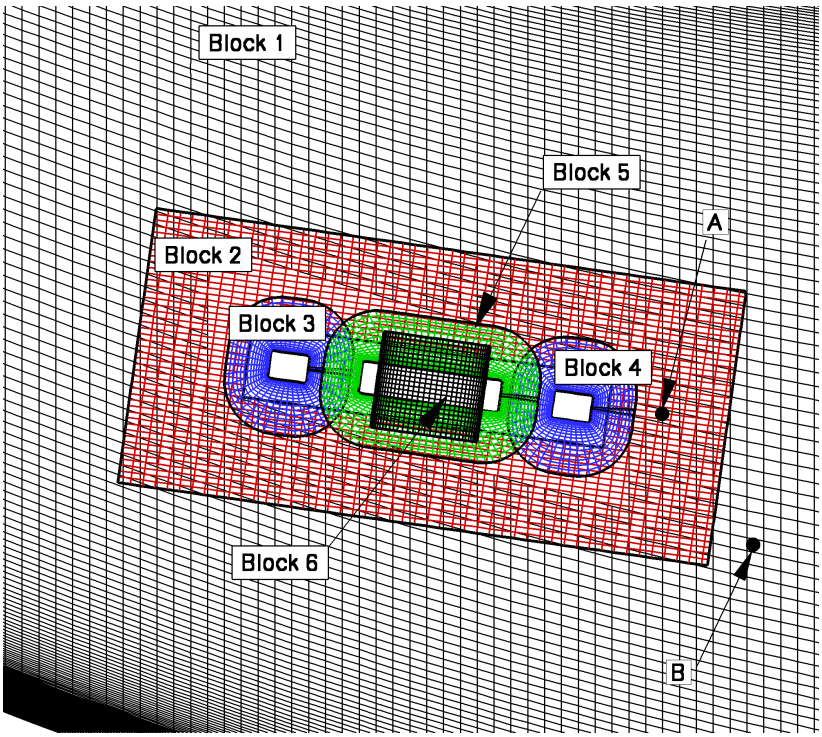


Figure 8: Overset grid for bridge foundation flow simulation

pseudo-time in each subdomain separately. During each physical time step, a converged solution for the entire flowfield is obtained by iterating in pseudo-time in each sub-domain and communicating boundary conditions at each boundary interface to connecting domain via interpolation. The interface interpolation is implemented in two steps. First we determine the location of each grid node on the interface relative to the other domains using a Newton method. Tang (2001) gives a detailed description of the grid-connectivity procedure. This grid connectivity information can be performed as a preprocessing step and stored for use during the calculation. The second step is the interpolation of the flow variables from the *donor domain* (which contains the background flow field information) to the interface of interest. Since we employ a second-order spatial discretization scheme for our flow solver, a second-order interpolation scheme is required to maintain the overall accuracy. The code we have developed features both standard trilinear interpolation for all flow variables at grid interfaces (Steger and Benek (1987)) as well as the so-called mass-flux based interpolation approach developed in Tang et al. (2003). This latter interpolation algorithm is based

on enforcing a second-order accurate discrete approximation of global mass conservation at each overset grid interface and has been shown to enhance the overall efficiency of the numerical algorithm as well as the smoothness of the computed solution across interfaces as compared to the standard tri-linear approach, especially when the grid spacing of adjacent subdomains is discontinuous. The details of the interpolation scheme and a summary of the entire iterative algorithm can be found in Tang et al. (2003).

Note that in the work of Tang et al. (2003) the governing equations were solved using an explicit (Runge-Kutta) dual time-stepping artificial compressibility iteration scheme. Tang et al. (2003) were able to obtain good results with this approach for a number of complex, low Reynolds number, laminar flows but our early attempts to extend their explicit iterative scheme to URANS simulations were not successful. Extensive numerical tests showed that when applying this method to solve the URANS and turbulence closure equations, numerical instabilities develop at grid interfaces during the early stages of the dual time-stepping iteration scheme. These instabilities were linked to the production terms in the turbulence closure equations, which depend on products of velocity gradients. Such terms impose more stringent than in laminar flow simulations smoothness requirements for both the velocity field and its gradients across grid interfaces. In fact we found that in order to avoid numerical instabilities during the early stages of the simulation, a smooth velocity field across interfaces needs to be established within as few dual time stepping iterations as possible. The more robust approximate factorization method we employ in this work was found sufficient for alleviating this problem and for allowing efficient and stable unsteady simulations over long time intervals.

2.4 Parallelization

The URANS solver described above solves the Navier–Stokes equations in a time–accurate manner on grids typically consisting of $O\left(10^6\right)$ nodes. A typical run of this unsteady flow solver requires more than 3000 thousand real time steps to obtain statistically converged solutions. Thus, a practical computation necessitates the use of powerful computational facilities. Parallel computers are ideally suited for such applications and provide a feasible

and inexpensive solution for such large–scale numerical simulation. However, the computer code must be parallelized to fully exploit the power of a parallel computer system. In this study, we parallelized our numerical solver on a SGI Origin 3800 system. The Origin 3800 system has 32 600MHz processors and is a shared–memory parallel system (SMP), which means that the memory installed in this system is accessible to all 32 processors. Such a computer structure allows the employment of the new emerging parallel mode, OpenMP, for parallelizing the numerical solver. OpenMP is targeted at shared memory parallel systems. The major advantage of OpenMP is its user friendliness and portability. In OpenMP, parallelism is expressed with a collective of compiler directives in the program thus leaving the original program structure untouched. Such a property requires minimum work on the original program to obtain increased performance on the numerical solver. Moreover, the OpenMP standard is ported to nearly every available platform thus the developed software can be easily transferred to other computer systems. In the following section, a brief introduction to OpenMP will be presented.

OpenMP is an Applicant Program Interface (API) that may be used for multi–threading programming. OpenMP uses fork–join model for parallel execution. First, the user specifies a number of parallel regions, where parallel execution is allowed, in the code by using the OpenMP API. The program starts as a single process: the master thread and runs sequentially until a parallel region is meet. Once entering into a parallel region, a number of additional slave threads are launched to create a team of threads for parallel execution. The master thread is a member of the team. The code within the parallel region is executed in parallel among the multi–threads in the team. On the end of each parallel region, all the threads in the team synchronize and terminate, leaving only the master thread. The program will run in the sequential mode again, until another parallel region is meet. This fork–join parallelism model is illustrated in Fig. 5. Although other forms of parallelization is also supported, OpenMP is mostly used to parallelize loops. We can insert OpenMP API into the original sequentially programmed code, thus telling the compiler that the loops between these specific APIs will be executed in a parallel manner. This parallelization method allows the programmer to reuse almost the entire sequential code base without any

major change, thus reducing the parallelization work load to the minimum. An efficiently parallelized code can be obtained within relatively short time given an available shared memory parallel computer system.

The OpenMP API has three major components:

- Compiler directives
- Runtime library routines
- Environmental variables

Compiler directives are a set of predefined directives that can be inserted into source code to tell the compiler how the code is going to be parallelized. The runtime library routines and environmental variables can be used to control and query the parallel execution environment. The OpenMP specifications are available both for C/C++ and Fortran programming languages. The current version of OpenMP specification is OpenMP 2.0 for all these languages.

The following example shows a typical do loop parallelized with OpenMP. The code is written in Fortran 90.

```
!$OMP PARALLEL PRIVATE(i,j,k) SHARED(im,jm,km,ucn,csi,eta,zet,q)
!$OMP DO
do k = 1,km;   do j = 1,jm;   do i = 1,im
      ucn(1,i,j,k) = csi(1,i,j,k) * q(2,i,j,k) +   &
           &  csi(2,i,j,k) * q(3,i,j,k) + csi(3,i,j,k) * q(4,i,j,k)
      ucn(2,i,j,k) = eta(1,i,j,k) * q(2,i,j,k) +   &
           &  eta(2,i,j,k) * q(3,i,j,k) + eta(3,i,j,k) * q(4,i,j,k)
      ucn(3,i,j,k) = zet(1,i,j,k) * q(2,i,j,k) +   &
           &  zet(2,i,j,k) * q(3,i,j,k) + zet(3,i,j,k) * q(4,i,j,k)
end do;   end do;    end do
!$OMP END DO
!$OMP END PARALLEL
```

In the above example, the directive started with *!$OMP* is inserted to specify the parallel section. The code chunk between *!$OMP Parallel* and *!$OMP End Parallel* will be implemented by the compiler to execute in a parallel mode. The clause after *!$OMP Parallel* is used to specify the properties, private or shared, of the variables appeared in the parallel code chunk. More detail information about the OpenMP API can be found on http://www.openmp.org.

The most important measure of parallelization is the scalability of the code, i.e. the speedup performance with increased number of processors. Speedup is measured as $T(1)/T(n)$ where $T(n)$ is the execution time of the code on n processors. In order to measure the scalability, a 3D driven–cavity is selected as a model problem. Three different grid levels, which are 32^3, 64^3 and 128^3 respectively, are designed to assess the code scalability under different grid dimensions. Fig. 9 shows the measured speedup for these three different grid levels. As can be seen in this figure, continuously speedup were obtained with increased number of processors. The obtained speedup of 8 processors for medium and larger size problems is around 6, which is comparable with the performance of the state-of-the-art commercial CFD software such as Fluent (e.g., Kremenetsky et al., 2000). The parallel performance is consistent under different grid dimensions, with a slightly better performance at larger grid dimensions.

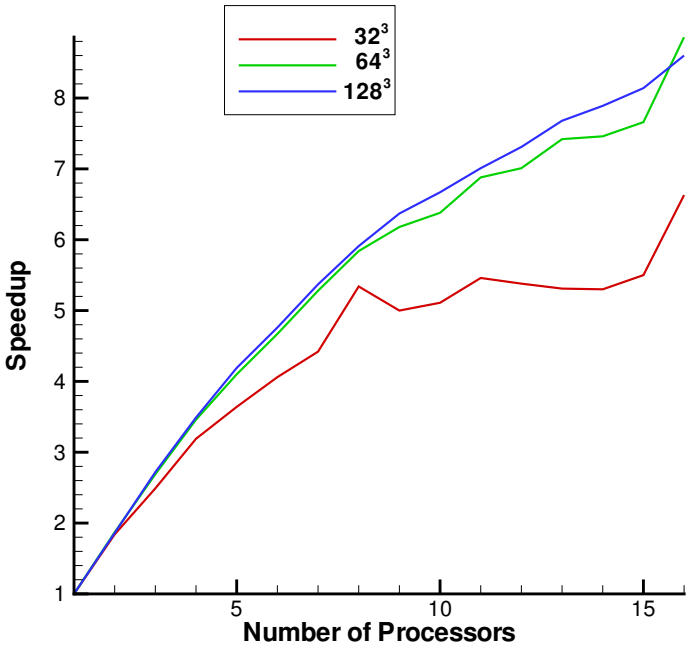


Figure 9: Speedup performance of the parallelized code

CHAPTER III

FLOWS AROUND BRIDGE PIERS

Over 650,000 bridges exist in the United States (FHWA 2001) and they constitute an important part of the country's transportation infrastructure. Therefore, and in addition to the potential for loss of life, bridge failure could have a dramatic impact on local economy. During the past several decades, several catastrophic scour induced bridge failures have occurred and up until today bridge scour remains the leading cause of bridge failures. The collapse of the Thruway Bridge over the Schoharie Creek on the New York State in 1987 during a near–record flood, during which 10 people died, attracted national attention to bridge scour problems in the US. In a recent study conducted by Wardhana and Hadipriono (2003), it was found that more than half of all bridge failures occurred between 1989 and 2000 in this country were the result of scour related damage.

By definition, bridge scour is the " erosion or removal of streambed or bank material from bridge foundations due to flowing water, usually considered as long–term bed degradation, contraction and local scour" (Federal Highway Administration). The removal of bed material is a direct result of the complex fluid dynamics near the bridge foundation. Bridge foundations flows are highly turbulent and characterized by three-dimensional separation, vortex formation, and large-scale unsteadiness. Unsteady coherent structures such as horseshoe, tornado, and whirlpool-like vortices have been identified in laboratory and numerical investigations of such flows (Dargahi, 1990; Chrisohoides et al., 2003) and their combined action has been linked to the initiation of bed erosion and scouring. Dargahi (1989, 1990) conducted a series of clear water scour experiments around a cylindrical pier using a hydrogen bubble flow visualization technique. These experiments revealed an intricate web of highly unsteady large-scale vortices in the upstream and downstream regions of the cylindrical pier. By carefully monitoring and observing the flow in the vicinity of the

scour hole, Dargahi was able to show that these vortices were indeed responsible for initiating bed erosion and scouring processes. Chrisohoides et al. (2003) reported laboratory visualizations and 3D, unsteady RANS, numerical computations for flow near a flat-bed abutment. Their results revealed a very complex and highly unsteady system of coherent vortices both in the upstream recirculating flow and the shear layer emanating from the abutment edge. By comparing the simulated three-dimensional flow structures with laboratory scour experiments, Chrisohoides et al. (2003) concluded that the growth of the scour hole in this case is enhanced by a downward velocity component in the vicinity of the pocket of maximum bed shear stress.

These studies clearly suggest that a critical prerequisite for the development of a quantitatively accurate numerical model of bed erosion and scour is the development of a fully 3D, unsteady hydrodynamic model capable of simulating the rich dynamics of coherent vortices at bridge foundations. Numerical investigations of such flows have been rather limited so far and for the most part have been restricted to steady RANS computations. Olsen and Kjellesvig (1998) developed a hydrodynamic model coupled with a sediment transport model to simulate the flow and scour around a single cylindrical pier. They obtained promising results but due to the steady nature of the simulation and the rather coarse mesh they employed, their simulations could not capture the unsteadiness in the flow. Tseng et al. (2000) studied the flow around square and circular piers with LES, which resolves the unsteady nature of the flow. Their simulation, however, was conducted on a rather coarse grid system with approximately 8×10^4 grid nodes. Given the complexity and high Reynolds number of this flow problem, accurate LES would require computational grids with at least 10^7 nodes (Spalart, 2000) and would be too expensive to carry out with available computer resources. Ali and Karim (2002) employed commercial software FLUENT to investigate the flow around a circular cylinder. For computational expedience, however, they assumed that the flow was steady and symmetric about the geometrical plane of symmetry and, thus, did not resolve any unsteady features.

In this chapter, we apply the numerical solver developed in this research to simulate the complex flow patterns in the vicinity of real-life bridge piers. A portion of an actual

bridge foundation located in the Chattahoochee River near Cornelia, GA is selected for the simulations and calculations are carried out for the following three different geometrical configurations: a) a single pier bent mounted on the actual river bathymetry; b) a single pier bent mounted on a flat bed; and c) two pier bents mounted on a flat bed. The first setting is selected to illustrate the capabilities of the developed numerical solver to handle the geometrical complexities of natural river reaches and in particular the large disparity in scales between the river reach and the bridge piers at real-life Reynolds numbers. The second configuration is selected to model the laboratory-scale configuration, which was employed to collect mean velocity and turbulence kinetic energy data to validate the numerical solver. The third configuration is used to demonstrate the potential of the method to simulate unsteady, coherent vortex shedding past complete bridge sections.

3.1 Flow past a pier bent mounted on the bed of Chattahoochee River

To demonstrate the capabilities of our numerical method to resolve complex, coherent, vortex shedding at the scale of the foundation while accounting for the large-scale bathymetric features of the reach in which the foundation is embedded for flows at real-life Reynolds numbers, we apply it to calculate turbulent flow through the Chattahoochee River reach shown in Fig. 7. The composite domain discretization shown in Figs. 7, 8 and 10 and discussed in Chapter 2 is employed for the simulation. The total number of active grid nodes for all six subdomains is approximately equal to 1.4×10^6. The calculations are carried for flow parameters that are typical to the specific reach. The upstream mean bulk velocity is $U_0 \simeq 2.1 m/s$, which leads to a Reynolds number $\Re = 2.37 \times 10^7$ based on the upstream bulk velocity and the width of the first bridge pier. Boundary conditions are specified as described in Chapter 2.3. The standard $k-\varepsilon$ model with wall functions is used to close the URANS equations.

The non-dimensional physical time step is set equal to $\Delta t = 0.25$ and approximately 20 to 30 dual time stepping iterations are required per physical time step to reduce the residuals by three orders of magnitude. Numerical experiments showed that a simulation

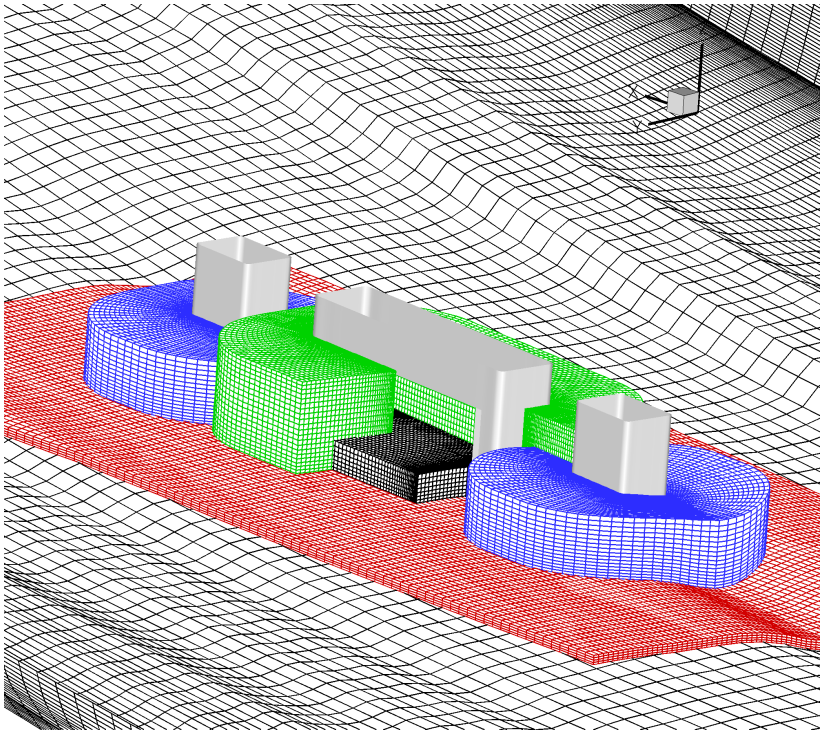


Figure 10: 3D view of the overset grid system.

interval of approximately 3000 physical times is adequate for obtaining statistically converged solutions. Completing such a simulation requires about 3 days of CPU time on 8 SGI Origin 3800 processors.

As discussed above, an important attribute of our method is that it is capable of resolving the unsteady vortex shedding in the vicinity of the bridge piers while accounting for the large-scale topography features of the overall reach. To illustrate this point we show in Fig. 11 calculated time histories of the transverse velocity component along y direction at two points in the flow. Point A is located in Subdomain 6 just downstream of the last pier while Point B is located in the background grid (Subdomain 1)—for point locations see Fig. 10 a). As seen in Fig. 11, following an initial transient of approximately 200 time units, the flow at Point A attains an oscillatory, periodic state. The transverse velocity fluctuates about a mean value of approximately 0.05 between 0 and 0.1 and the intensity of the oscillations is sustained throughout the entire simulated interval. In stark contrast, the flow at Point B reaches a quasi-steady state after approximately 200 time units—a

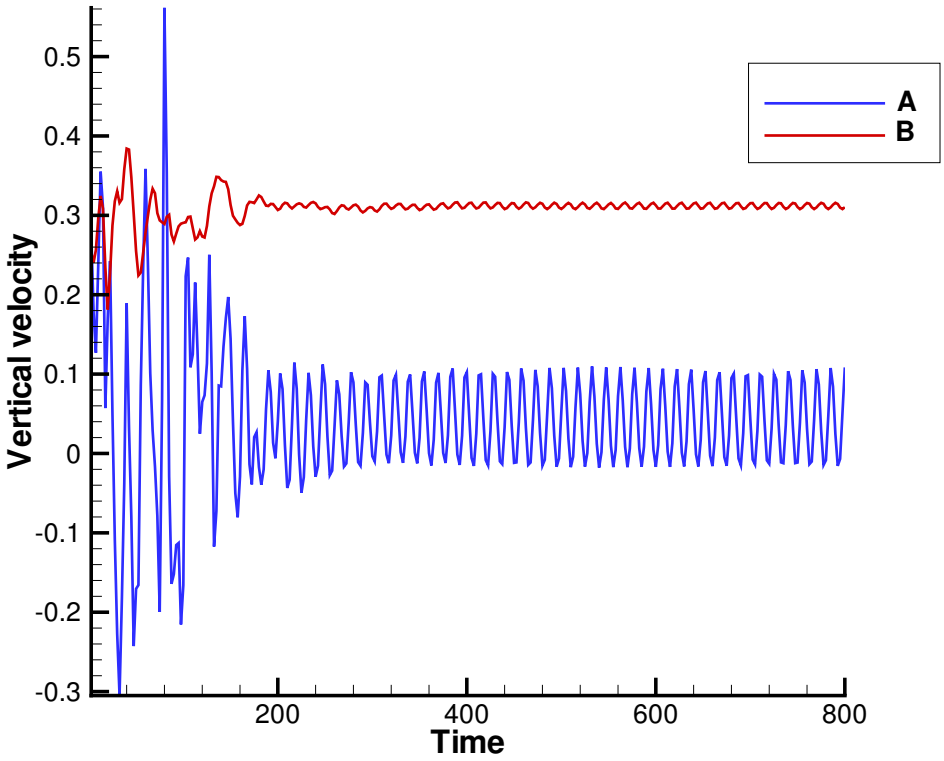


Figure 11: Timehistory of vertical velocity component at points A & B (see Fig. 8 for detail locations).

very weak unsteady fluctuation persists at this point but its amplitude is only a very small fraction of the mean vertical velocity at Point B. This trend is consistent with the resolution of the background grid, which is far too coarse to sustain the unsteadiness generated in the finer grids embedded around the structures. The role of this grid is to resolve the important features of the approach and ambient flow and, thus, help define realistic inflow, outflow and farfield boundary conditions for the flow in the vicinity of the pier bent. Therefore, the results shown in Fig. 11 demonstrate clearly the ability of our method to resolve coherent vortex shedding in the vicinity of the piers using a locally refined mesh distribution.

To illustrate the complexity of the large-scale flow in the vicinity of the piers we show in Fig. 12 two snapshots of instantaneous velocity magnitude contours and streamlines at a horizontal plane located just below the horizontal slab connecting the two middle piers.

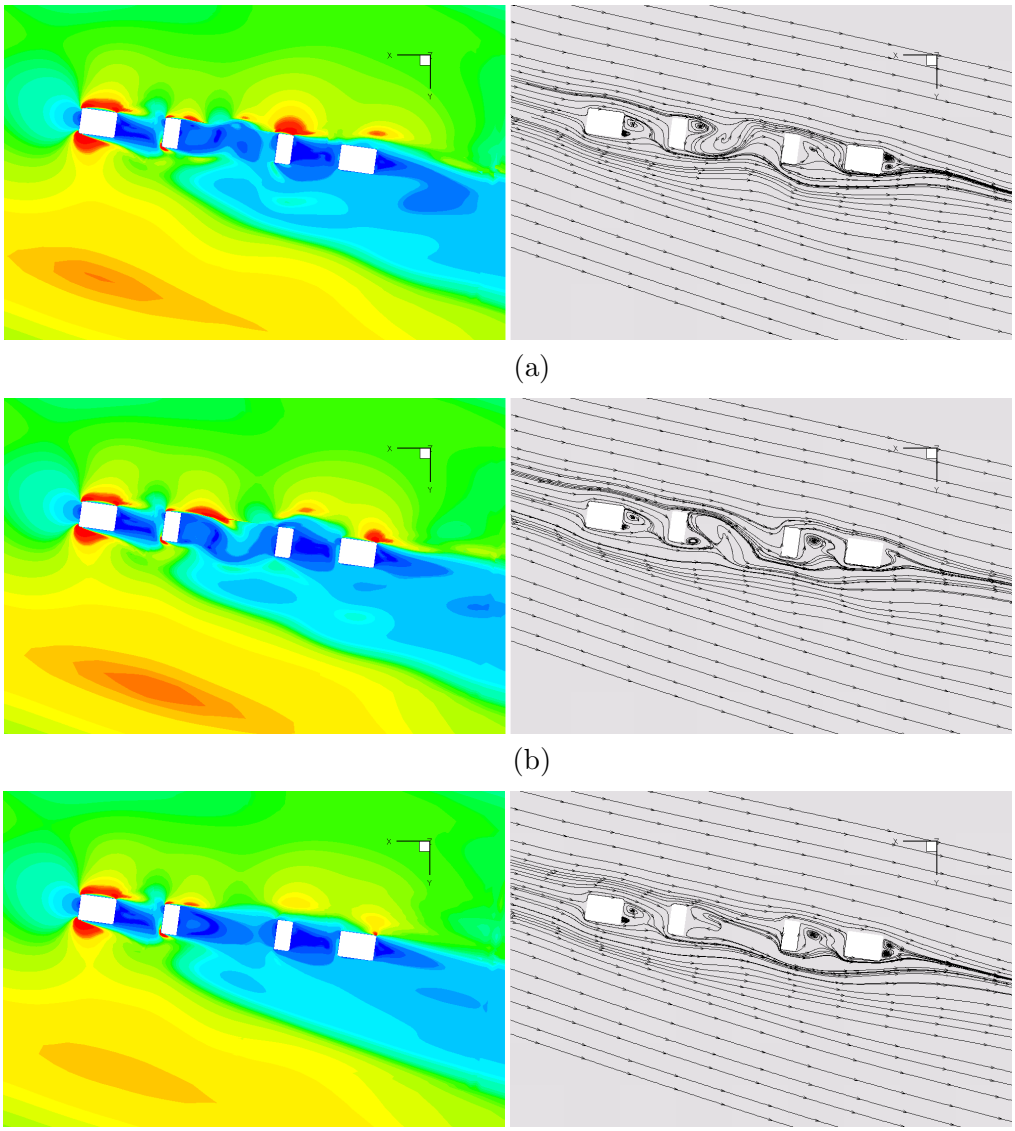
The corresponding time-averaged image is also included for reference–the time averaged flow is obtained by averaging the calculated instantaneous flowfields over the simulated 3000 time steps. This figure illustrates clearly the complexity of the flow in the vicinity of the piers. Features such as the unsteady meandering of the shear layers around the bent and the unsteadiness of the recirculating flow regions in the pier wakes are clearly evident in these results. The effect of the complex river bathymetry on the pier hydraulics is also apparent. Note for instance that both the instantaneous and time-averaged flowfields are highly asymmetric. The velocities are considerably higher and the vortex shedding is more intense on the left (as viewed looking toward the piers from upstream) side of the pier bent due to the fact that the approach flow is skewed by about 5° relative to the streamwise axis of the bent. Obviously such complex flow patterns can not be accurately simulated without taking into account the complexity of the ambient bathymetry. The ability of the method to capture very complex, 3D flow patterns in the vicinity of the piers is further illustrated in Fig. 13, which depicts snapshots of instantaneous particle paths. Highly unsteady, coherent vortical structures with axes perpendicular and parallel to the bed are seen to appear and disappear continuously throughout one period of the flow.

Finally, it is important to emphasize that the simulated unsteadiness is not the result of any unsteady forcing imposed artificially on the inlet flow. Unsteady vortex shedding is rather naturally excited as a result of large-scale instabilities of the flow induced by the presence of the piers. The ability of our method to capture and sustain over long simulated time intervals such dynamically rich phenomena in such a complex geometrical configuration is a testament to its high spatial and temporal resolution and underscores its promise as a powerful simulation tool for real-life hydraulic engineering flows.

3.2 Flow past a pier bent on flat bed

3.2.1 Computational Details and Overview of Simulated Cases

In this section we report detailed numerical sensitivity studies and validation of the flow solver developed in this work. Calculations are carried out for flow past the same pier bent studied in the previous section mounted on a flat bed. Mean flow and turbulence kinetic

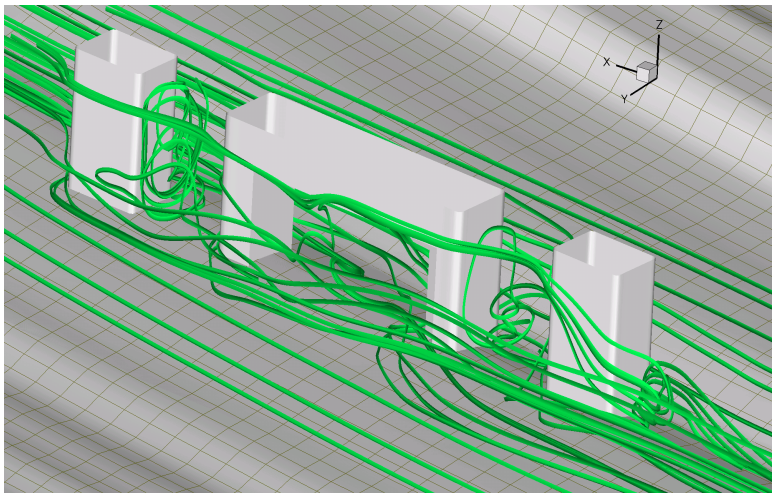


(a)

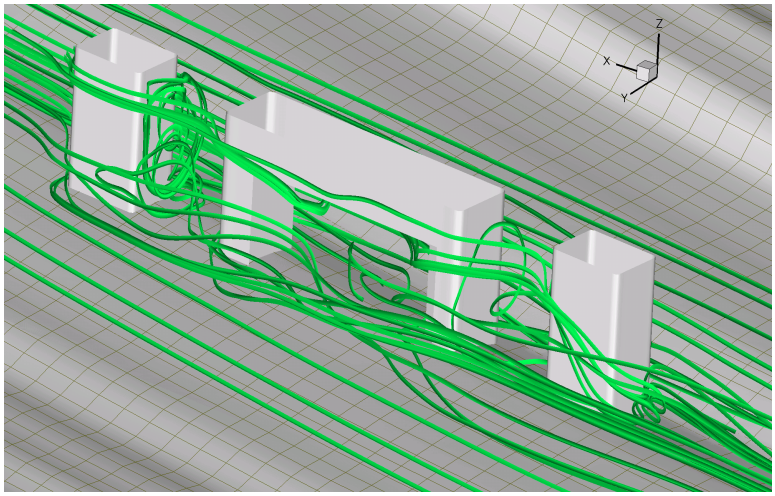
(b)

(c)

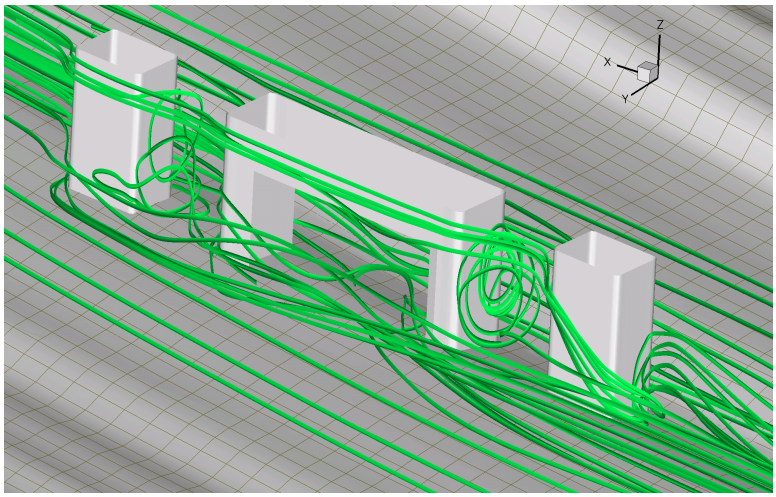
Figure 12: Contours of horizontal velocity magnitude (left) and limited streamtraces (right) or a horizontal plane. (a) & (b) are instantaneous flow field of two different time instants, (c) shows the properties of average flow field.



(a)



(b)



(c)

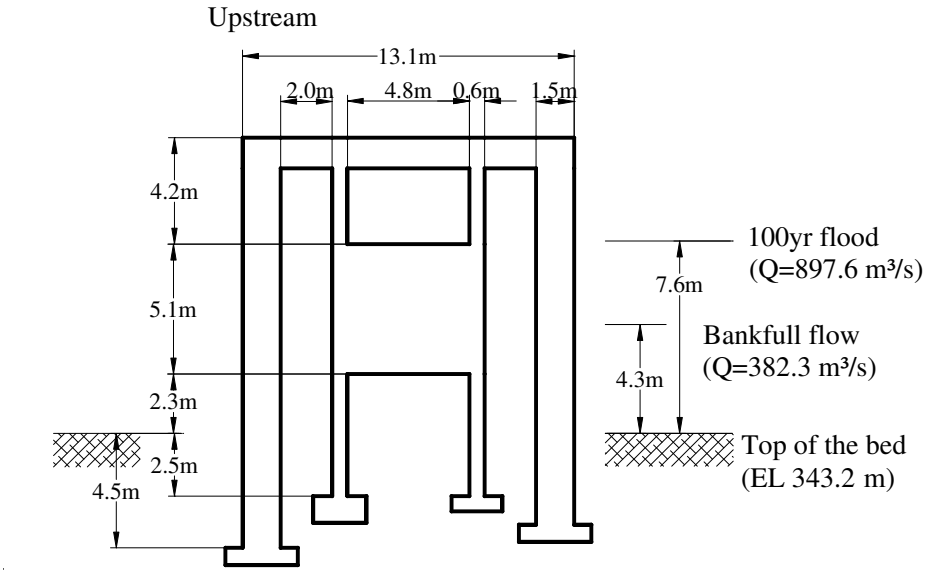
Figure 13: 3D instantaneous stream traces at three different time instants.

measurements for this configuration were obtained at the scour flume of the Hydraulic Laboratory at the Georgia Institute of Technology. The details of the experimental set up and measurement techniques are described in Sturm et al. (2004) and Ge et al. (2004).

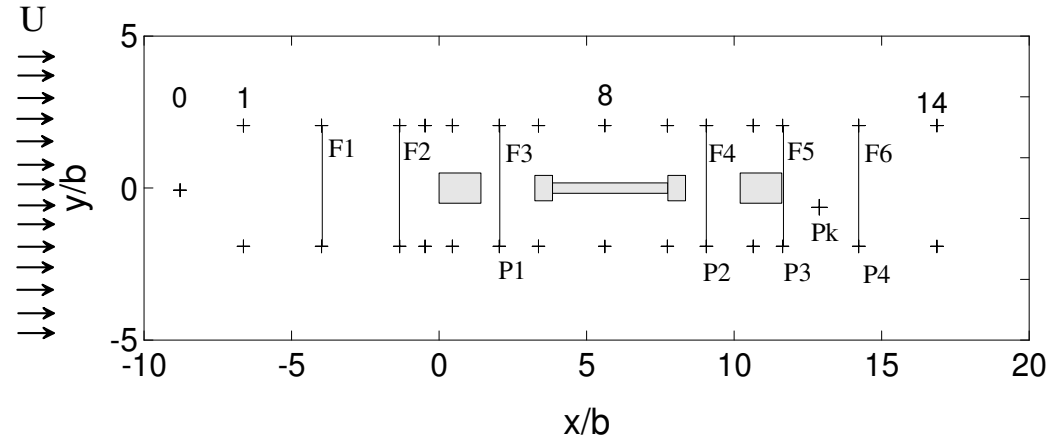
The geometry of the pier bent with the prototype dimensions and the measurement locations are shown in Fig. 14. All calculations were carried out for Reynolds number $Re = 28516$, based on the upstream bulk velocity and the width b of the first bridge pier, which corresponds exactly to the experimental conditions. As in the previous section, the standard $k - \varepsilon$ model with wall functions is used to close the URANS equations. In our subsequent discussion we shall refer to the four piers of the foundation as pier 1 – 4, with pier 1 corresponding to the first pier facing the upstream flow and the remaining piers numbered along the direction of the flow.

The computational domain is a rectangular box with the streamwise axis of the pier bent coinciding with the streamwise axis of symmetry of the box. The upstream boundary of the domain is placed $7b$ upstream of the center of pier 1 and the overall streamwise length of the domain is $34b$. The lateral dimension of the computational domain is $14b$ and the flow depth is $4.16b$. The computational domain is decomposed into 5 overset subdomains and each subdomain is discretized with a separate body-fitted curvilinear mesh as shown in Fig. 15. To investigate the sensitivity of the computed solutions to grid refinement, we carry out computations on two grids: a coarse grid with a total of 8×10^5 nodes and a fine grid with a total of 1.6×10^6 nodes. For both grids the same 5-subdomain decomposition and the same dimensions of the computational box ($34b \times 14b \times 4.16b$) are used. For the sake of clarity in the remainder of this paper we shall refer to the coarse and fine grids as cases C1 and C2, respectively.

To investigate and quantify any possible effects that the application of boundary conditions at a truncated, relative to the experimental, computational domain may have on the near foundation flow patterns, we also carried out a simulation for a wider computational box having dimensions of $34b \times 16b \times 4.16b$. This domain was discretized with the same 5-subdomain overset grid layout and with the coarse 8×10^5-node mesh. We shall refer to this case as case C3.



(a)



(b)

Figure 14: (a). Profile view of central pier bent with prototype elevations and dimensions; (b) Plan view of central bridge pier bent and locations of mean flow and turbulence measurements. (b_0 = width of the prototype upstream pier = 1.07m). Lines marked F1 to F6 indicate locations where measured mean velocity profiles in the transverse, y, direction are compared with measurements. Points marked P1 to P4 indicate locations where measured turbulence kinetic energy profiles in the vertical (depth) direction are compared with the measurements.

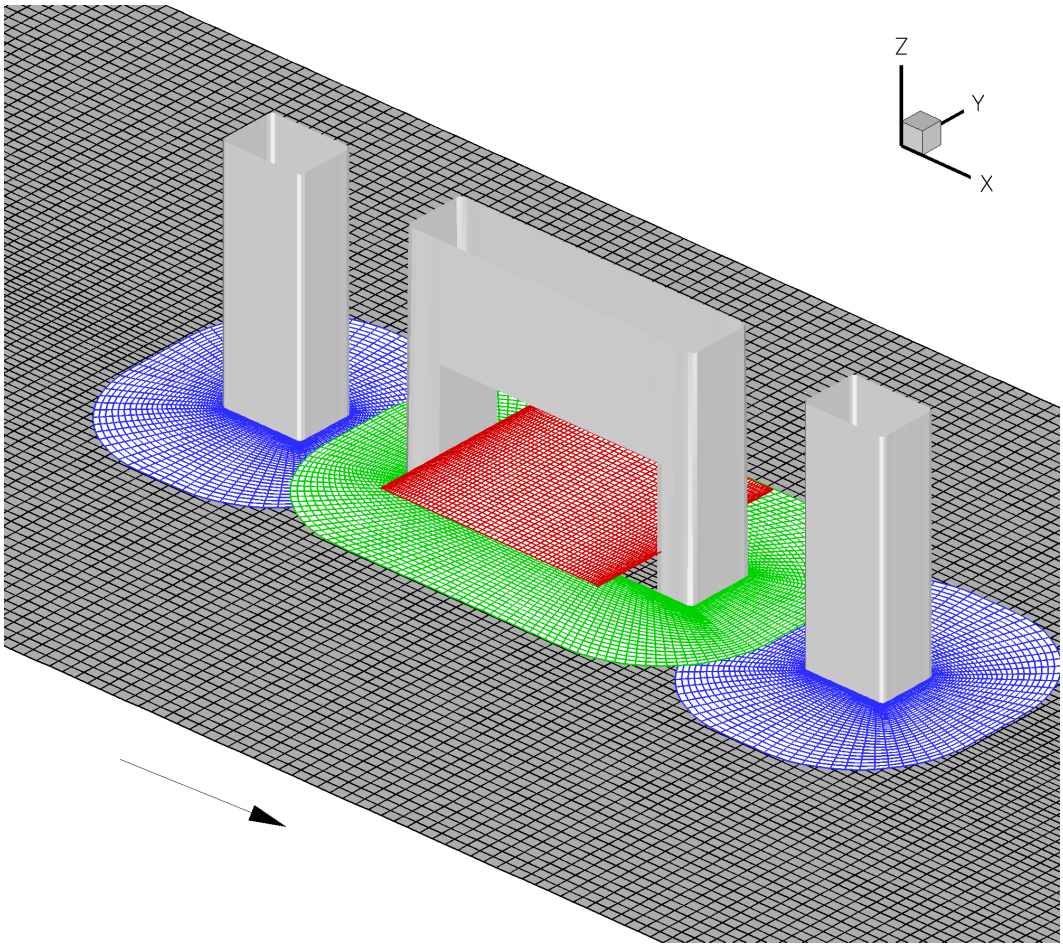


Figure 15: Computational domain and overset grid layout for single bent on flat river bed.

Table 1: Details of simulated cases

Case	Grid nodes	Domain size $x \times y \times z$	Flow alignment
C1	0.8×10^6	$34b \times 14b \times 4.16b$	Yes
C2	1.6×10^6	$34b \times 14b \times 4.16b$	Yes
C3	0.8×10^6	$34b \times 16b \times 4.16b$	Yes
C4	0.8×10^6	$34b \times 14b \times 4.16b$	No

The effect of approach flow alignment with the piers was investigated by carrying out yet another calculation for which the streamwise axes of the computational domain and the pier bent, respectively, were offset by a very small angle of 1.8°. This case will be referred to as case C4 and was discretized using exactly same grid arrangement as for case C1.

The same boundary conditions were employed for all simulated cases. At the inlet, a fully developed turbulent flow velocity profile is specified obtained from a separate straight-channel computation. The free surface is approximated as a flat rigid-lid. At the lateral boundaries and the outlet of the flow domain, flow variables are obtained using linear extrapolation from the interior of the domain. The generalized rough-wall functions approach described in Chapter 2.2.1 is employed to specify boundary conditions for the velocity components and turbulence quantities at the channel bottom.

For all cases the simulations were carried out using a non-dimensional time step of $\Delta\bar{t} = \Delta t/T = 0.2$ ($T = b/U_0$) as some test simulations with a lower time step showed no appreciable differences in the simulated flow fields. For each case, 5000 physical time steps were found more than adequate for obtaining statistically converged results. During each time step, the dual-time iteration procedure was declared converged when the residuals were reduced by three orders of magnitude, a convergence tolerance which typically required 20 to 30 pseudo-iterations per time step.

The details of all four simulated cases are summarized in Table 1.

3.2.2 Results and discussion

The presentation of the computed results and comparisons with the measurements are organized as follows. First, we provide an overview of the general flow patterns around the single pier bent configuration with emphasis on the characteristics of the large-scale unsteadiness and its contribution to the production of turbulence kinetic energy. Next, we validate the numerical model and investigate the sensitivity of the computed results to grid refinement, domain length, and upstream flow alignment by comparing mean velocity and turbulence kinetic energy profiles for cases C1, C2, C3, and C4 with experimental data. Finally we discuss the simulated three-dimensional flow patterns in relation to the observed equilibrium scour patterns in the laboratory to establish the link between complex hydrodynamics and sediment transport phenomena.

3.2.2.1 Description of unsteady flow patterns

The results discussed in this section are for case C1 but very similar unsteady flow patterns are obtained for cases C2 and C3. Fig. 16 shows calculated time series of the resolved transverse velocity component at two points in the wake of the piers. This figure illustrates the periodic nature of the large-scale flow, which is established following a short initial transient of approximately 100 time units. Before we discuss the detail flow patterns observed in our numerical simulation, it is necessary to show that the result obtained in our simulation is independent of the time–step used for the simulation. As stated earlier, we use a dimensionless time step of $\Delta t = 0.2T$. In order to show the time independence, we halve the time–step used for the current simulation to $\Delta t = 0.1T$ and conducted a same calculation of case C1. Fig. 17 shows the time history of the same velocity component at the same points as shown in Fig. 16. The calculated time histories for both time steps are very similar and the power spectrum analysis (Fig. 18) shows that both simulations yield essentially the same shedding frequency. These results suggest that the time–step used in our simulation is fine enough and the obtained results are independent of the time step resolution.

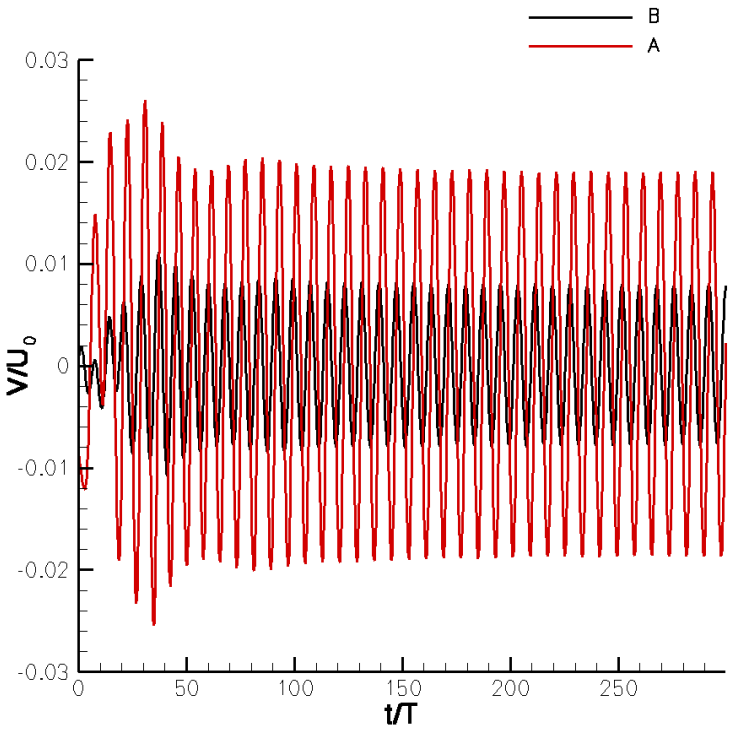


Figure 16: Calculated time histories of resolved v-velocity component at two different points in the wake of the foundation ($T = b/U_0$) for single bent on flat bed $dt = 0.2T$.

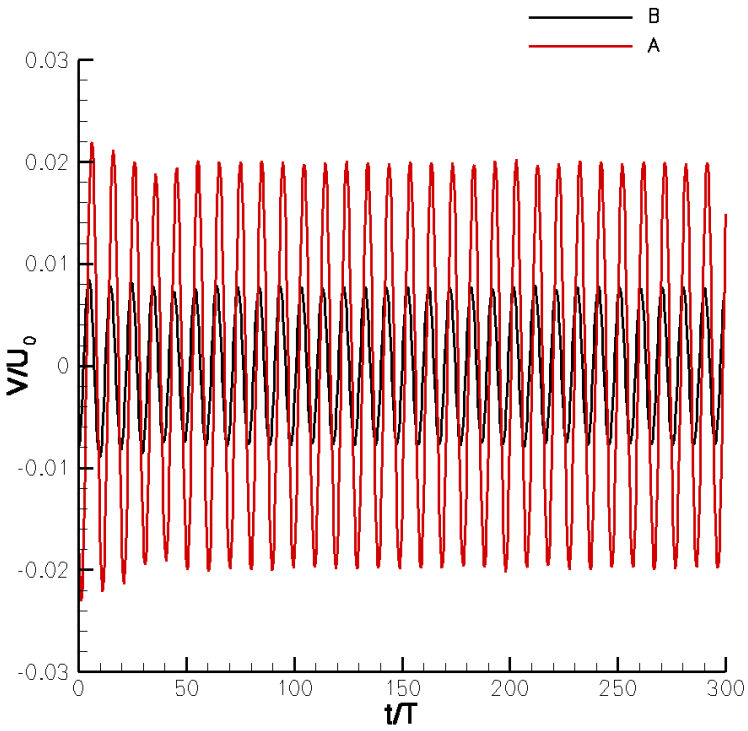


Figure 17: Calculated time histories of resolved v-velocity component at two different points in the wake of the foundation ($T = b/U_0$) for single bent on flat bed $dt = 0.1T$

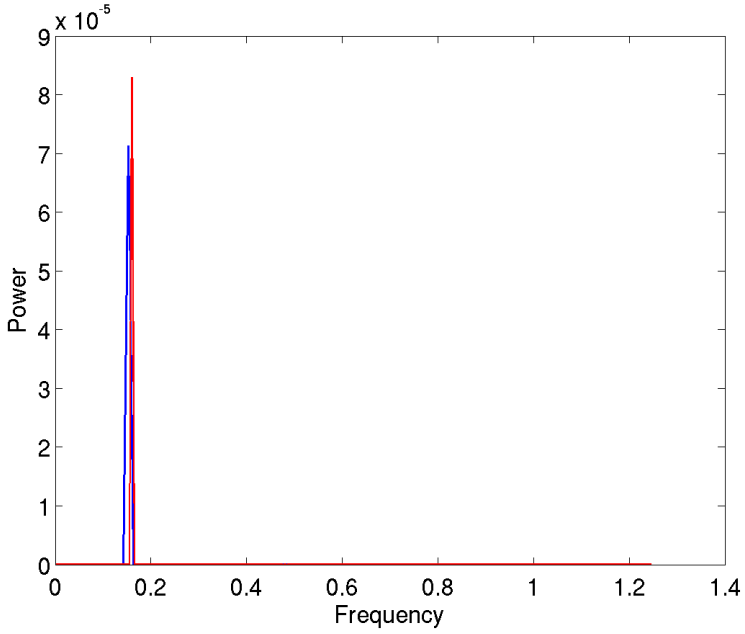


Figure 18: Power spectrum comparison of time step refinement

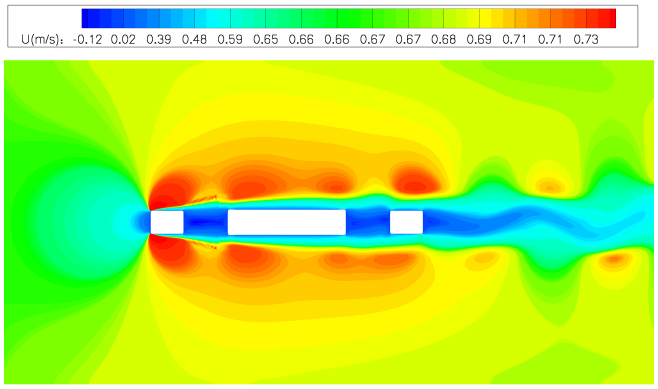
Juxtaposing the two snapshots of resolved streamwise velocity contours with the time-averaged flow at the horizontal plane shown in Fig. 19(a,b), clearly shows that unsteadiness in the flow originates due to the Kelvin-Helmholtz type instability of the shear layers emanating from the upstream corner of the foundation and the intense vortex shedding in the wake. The transverse flapping and meandering of the wake flow is clearly evident in the two snapshots in this figure. Furthermore, it should be noted that the time-averaged flow field shown in Fig. 19c exhibits a high degree of symmetry with respect to the streamwise axis of the foundation, thus, suggesting that the simulated time interval is sufficient for obtaining statistically converged mean flow.

To quantify the intensity of the resolved large-scale unsteadiness and its relative contribution to the total budget of the turbulence kinetic energy (TKE or k), we compare in Fig. 20 the contours of modeled and resolved k at the same plane as that shown in Fig. 19. Before we proceed with the interpretation of these plots, however, it is important to define the terms modeled and resolved TKE. In URANS simulations, the instantaneous flow variables are decomposed into three components: the time–averaged velocity $\overline{u_i}$, the phase–averaged velocity u_i'', and the turbulent fluctuations $u_i\prime$. By modeled TKE at a given point in the flowfield, k_m, we denote the time-average of the kinetic energy obtained from the time-accurate solution of the k-equation as follows:

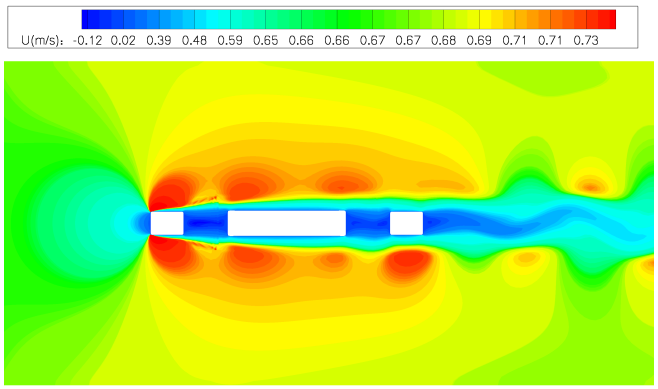
$$k_m\left(x,y,z\right) = \lim_{t\to\infty}\frac{1}{t}\int_0^t k\left(t',x,y,z\right)dt'$$

where $k\left(t',x,y,z\right)$ is the solution of the transport equation for k at point (x,y,z) at time t' with turbulent fluctuations relative to the phase–averaged velocity components. Therefore, k_m quantifies the amount of the turbulence kinetic energy modeled by the turbulence closure model. The resolved kinetic energy, k_r, on the other hand is the amount of energy due to the large-scale, coherent motions in the flow, which are resolved directly by solving the URANS equations. It is defined as follows:

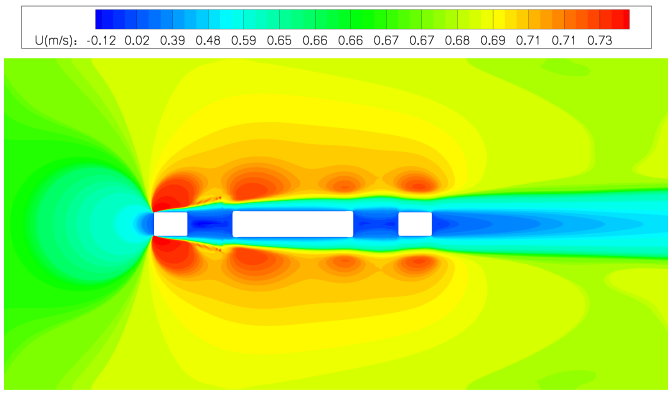
$$k_r = \lim_{t\to\infty}\frac{1}{t}\int_0^t \frac{1}{2}\left[u''^2+v''^2+w''^2\right]dt'$$



(a)



(b)



(c)

Figure 19: Instantaneous (a and b) and time-averaged (c) contours of streamwise velocity at $z = 0.7H$ from bottom.

where $u'' = u - \overline{u}$, u is the instantaneous, phase-averaged velocity component obtained from the solution of the URANS equations, and $\overline{u}$ is its time-averaged value. It is evident from Fig. 20 that the resolved kinetic energy produced by the coherent periodic vortex shedding accounts for a significant percentage of the total energy starting from the wake of the third pier. In fact in the wake of the foundation, k_r appears to overwhelm k_m. It is important to emphasize that the flow in the wake of the piers is highly three-dimensional, which leads to a significant variation with depth in the division of energy between modeled and resolved. This significant feature of the flow is depicted in Fig. 21, which compares vertical profiles of modeled and resolved kinetic energy at one point downstream of the piers (see Fig. 14(b) for point location). As seen in the figure the turbulence closure model accounts for most of the energy up to approximately fifty percent of the channel depth with this trend reversing as the water surface is approached. Even though not shown herein, we have found similar trends throughout the flow. Large–scale coherent shedding dominates in the upper half of the channel while the near–bed flow is dominated by smaller–scale incoherent motions, which are captured by the turbulence closure model.

Turbulence statistics from cases C2 and C3 are shown in Fig. 22 and Fig. 23 respectively. As seen in the figures, the results from these different cases are very close to each other, with small different distribution pattern shown in the solution of case C2, which has much finer grid resolution than the other two cases. All these figures along with the ones shown in Fig. 20 show that the solutions are independent on the grid resolution and boundary conditions used in this study.

3.2.2.2 Validation and grid/domain sensitivity studies

In Figs. 24 and 25 we compare measured and computed, for cases C1, C2, and C3, streamwise velocity profiles at various locations upstream, within, and downstream of the piers. The profiles show velocity variations in the transverse, y, direction at various streamwise locations and three different depths.

It is evident from Fig. 24 that upstream of the piers all three simulated cases are practically indistinguishable. Small discrepancies between the three numerical solutions are

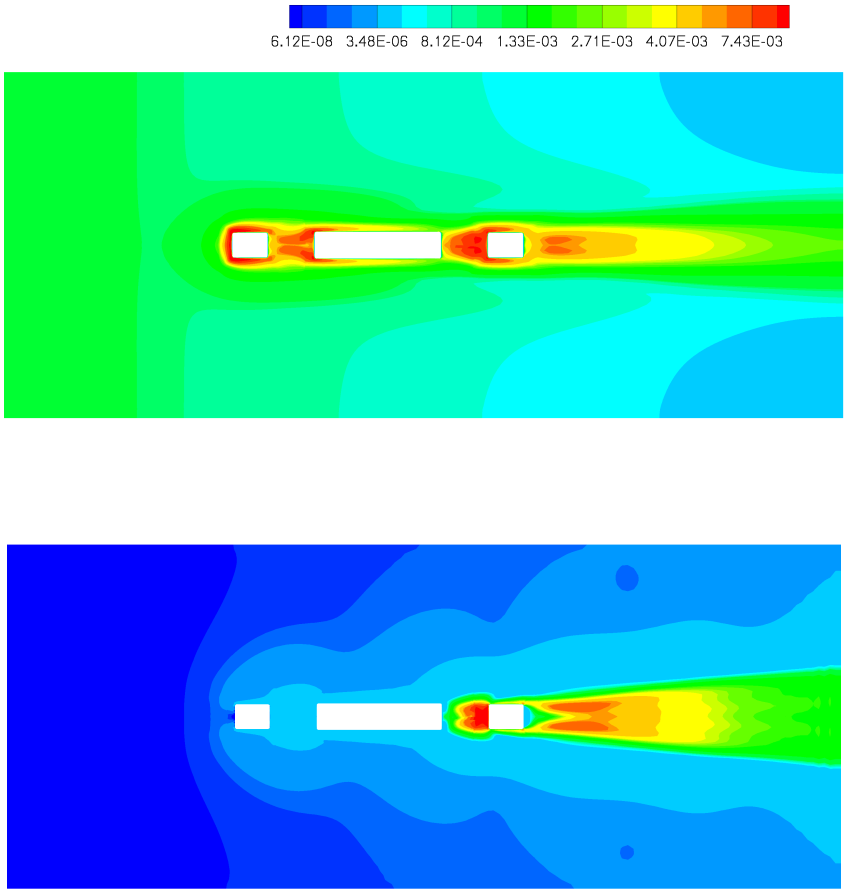


Figure 20: Distributions of turbulence kinetic energy at $z = 0.7H$ from bottom (Case C1). (top) modeled k_m/U_0^2 (bottom) resolved k_r/U_0^2

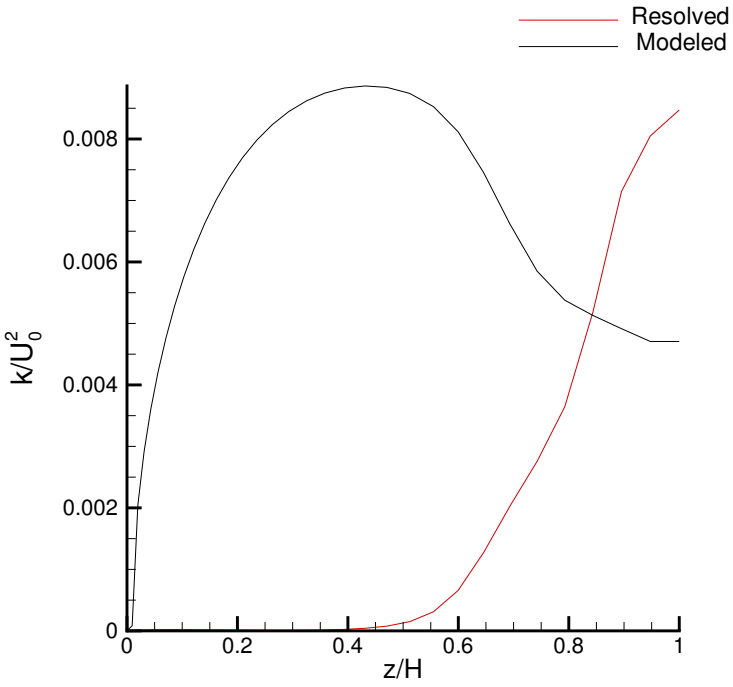


Figure 21: Calculated variation with depth of modeled (black) and resolved (red) turbulence kinetic energy at a point Pk downstream of pier 4.

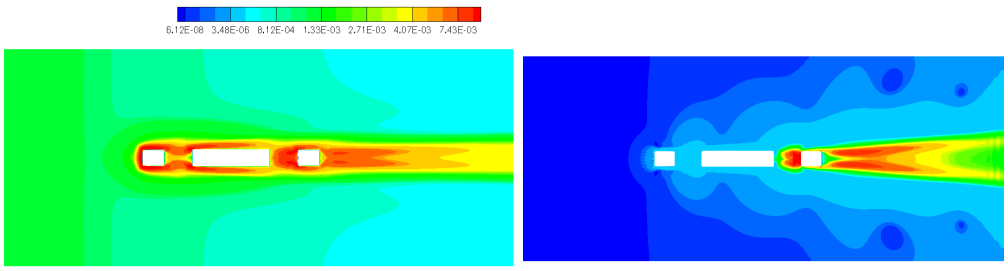


Figure 22: Distributions of turbulence kinetic energy at $z = 0.7H$ from bottom (Case C2). (top) modeled k_m/U_0^2 (bottom) resolved k_r/U_0^2

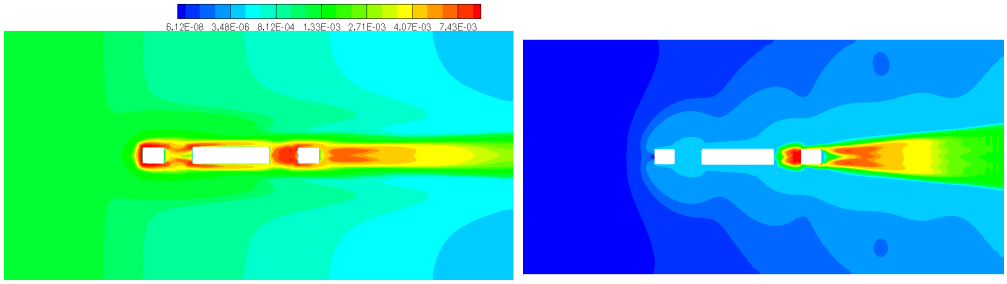
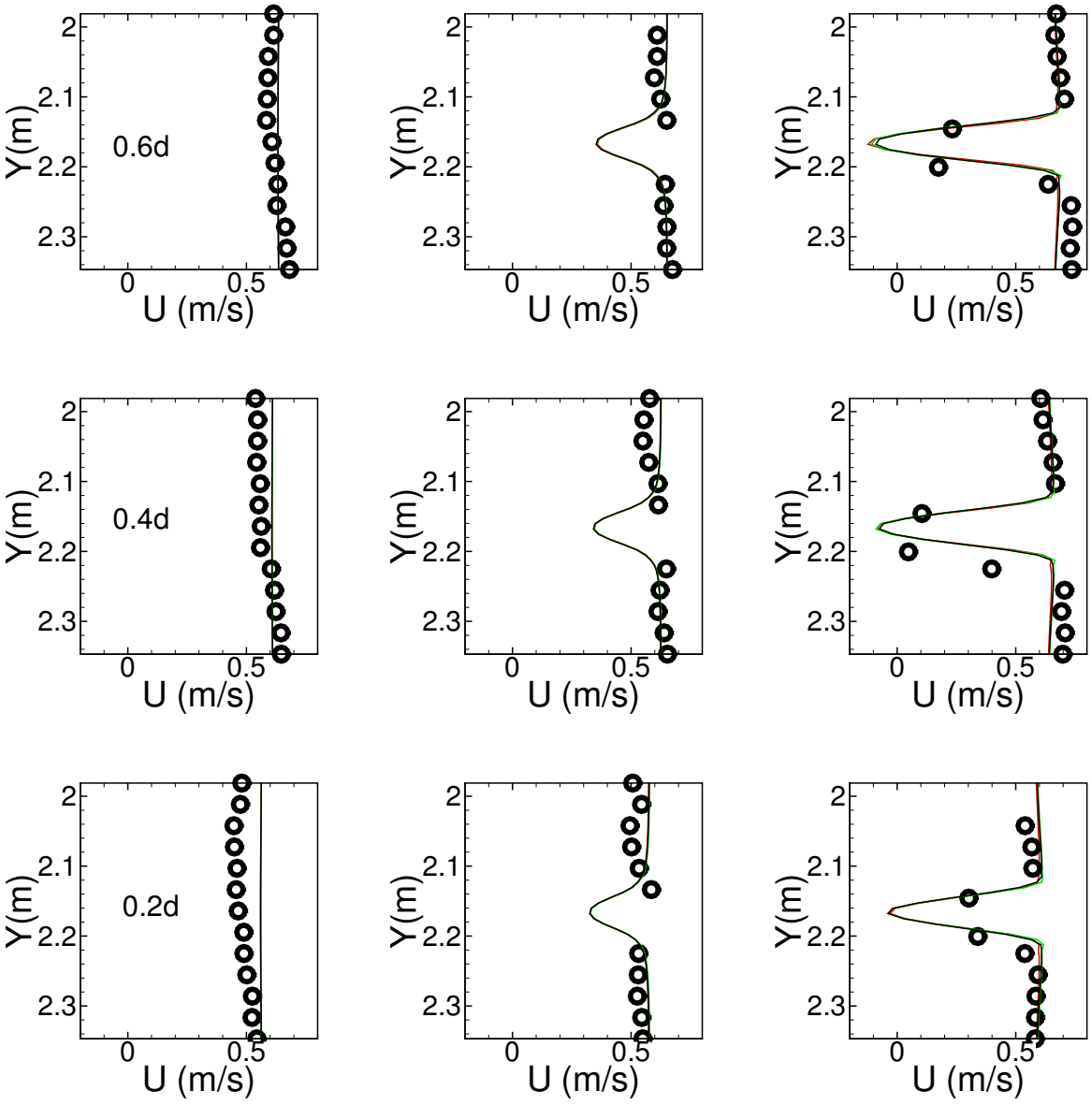


Figure 23: Distributions of turbulence kinetic energy at $z = 0.7H$ from bottom (Case C3). (top) modeled k_m/U_0^2 (bottom) resolved k_r/U_0^2

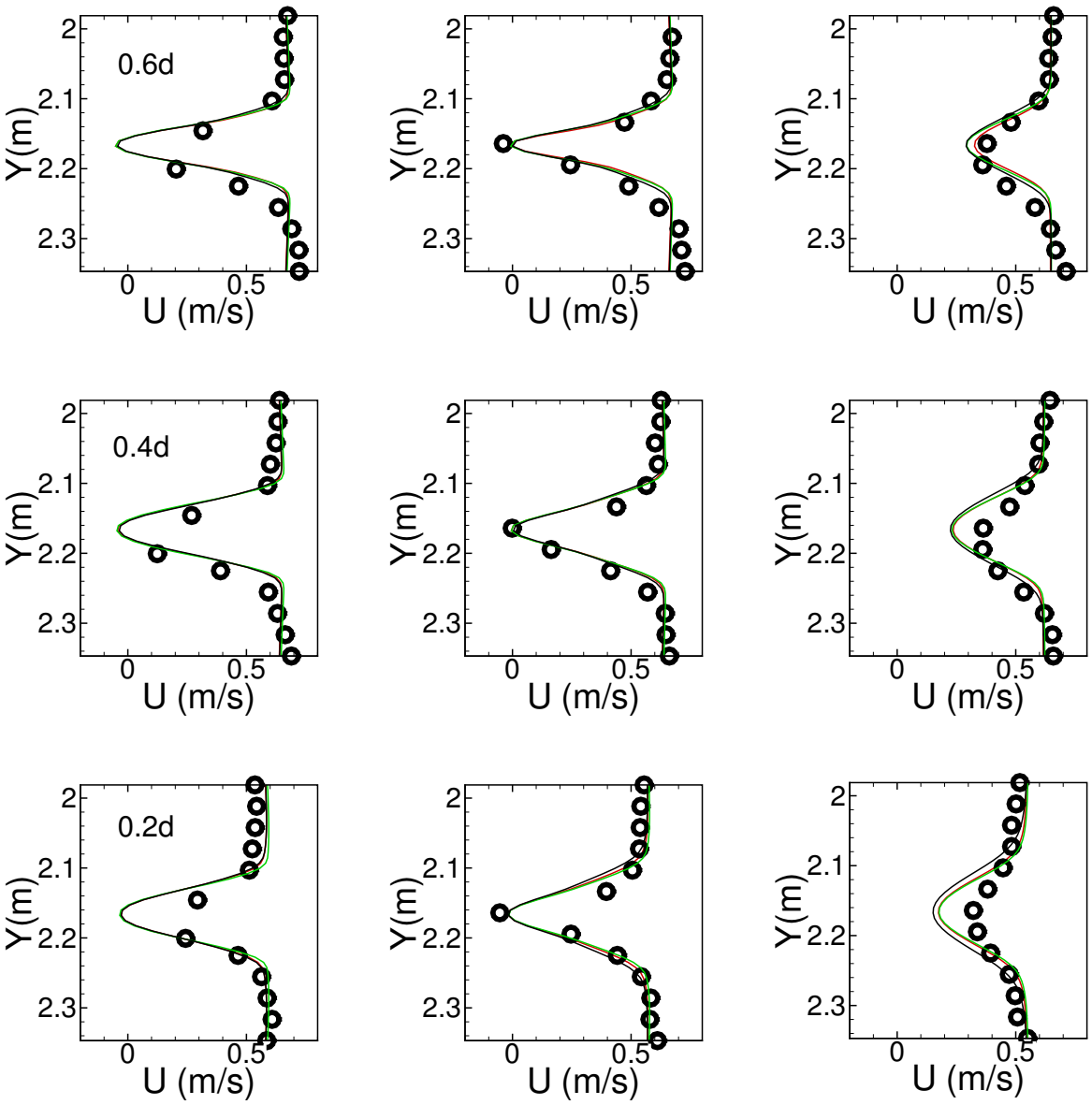
only observed in the wake of the piers, in Fig. 25. These discrepancies are visible mainly at the furthest downstream streamwise location F6, where the results obtained on the finest mesh (C2 case) yield somewhat larger velocity deficits in the wake. Overall, however, the coarse and fine mesh predictions are in excellent agreement with each other at all locations, which points to the conclusion that the coarse mesh is adequate for capturing the details of this flow. It also follows from this figure that the smaller computational domain used in cases C1 and C2 is sufficient for eliminating any spurious effects that the truncation of the flow domain and the specification of simplified boundary conditions at the lateral boundaries may have on the simulated flow structures near the piers.

The comparisons of the simulated and measured velocity profiles shown in Figs. 24 and 25 reveal that the numerical model captures most trends observed in the experiments with very good accuracy. For instance, both the reduction of the centerline velocity as the foundation is approached and the growth of the wake between and downstream of the piers are predicted with good accuracy by the numerical model. There are, however, several important points that need to be made concerning the comparisons shown in Figs. 24 and 25. First note that at the upstream most location (F1), Fig. 24, the measured velocity profiles are not perfectly symmetric and uniform. Moreover, the degree of non-uniformity in the measured profiles appears to vary with depth. On further investigation of the measured direction of the approach velocity vectors relative to the piers, it was discovered that there was a slight skewness of about 1.8° clockwise in the approach velocity



(a) F1 (b) F2 (c) F3

Figure 24: Comparisons of measured (open circles) and calculated streamwise mean velocity profiles in the transverse direction at various depths and streamwise locations upstream of the piers for cases C1 (red), C2 (green), and C3 (black). Streamwise locations: a) F1; b) F2; c) F3 (see Fig. 14(b) for measurements locations). Depth locations: From bottom to top, $0.2H$, $0.4H$ and $0.6H$ respectively.



(a) F4 (b) F5 (c) F6

Figure 25: Comparisons of measured (open circles) and calculated streamwise mean velocity profiles in the transverse direction at various depths and streamwise locations upstream of the piers for cases C1 (red), C2 (green), and C3 (black). Streamwise locations: a) F4; b) F5; c) F6 (see Fig. 14(b) for measurements locations). Depth locations: From bottom to top, $0.2H$, $0.4H$ and $0.6H$ respectively.

vector relative to the pier centerline (see Ge et al. (2004) for detail). This may have been due to inherent construction tolerances in setting the piers relative to the approach flow or due to a slight asymmetry in the approach flow itself. The asymmetry was not detectable in flow visualizations. We will subsequently show, however, that the flow field, and in particular the structure of turbulence near the foundation, is extremely sensitive to even the smallest degree of skewness, which is exacerbated by the solid interior pier web. These small but persistent variations of the upstream flow conditions in the experiment could not be accounted for in the numerical model since we have assumed fully-developed turbulent channel flow conditions at the inlet. The effect of this apparent asymmetry in the experimental model is evident in all measured profiles shown in Figs. 24 and 25. All measured profiles exhibit a slight asymmetric bias with respect to the streamwise axis of symmetry of the foundation and this trend is more pronounced in the wake region (see Fig. 25). This small deviation from symmetry notwithstanding, however, the simulations capture the growth and evolution of the wake with good accuracy. The most notable discrepancies between experiments and simulations are observed at the farthest downstream location in the wake (F6) near the bed where the measured wake profile suggests a somewhat faster than simulated rate of recovery of the wake flow.

To further verify our numerical simulation, we compare in Fig. 26 calculated and measured profiles of total turbulence kinetic energy ($k_t = k_r + k_m$) in the vertical (depth) direction at several streamwise locations to the left and right of the foundation (locations are marked as P1 to P4 in Fig. 14). Since all cases from C1 to C3 yield very similar solutions, only results from case C1 are included in this figure. A remarkable feature of the measured kinetic energy profiles is the large asymmetry of the turbulence structure with respect to the streamwise axis of symmetry of the foundation. At the first location (P1) the measured kinetic energy profiles to the right and left of the foundation are nearly identical and in good agreement with the simulations. Further downstream, however, the kinetic energy to the right side of the foundation rises sharply in the outer layer yielding a highly asymmetric turbulence structure. The numerical simulations, on the other hand, yield a symmetric turbulence structure, which is to be expected since the simulated conditions are

perfectly symmetric.

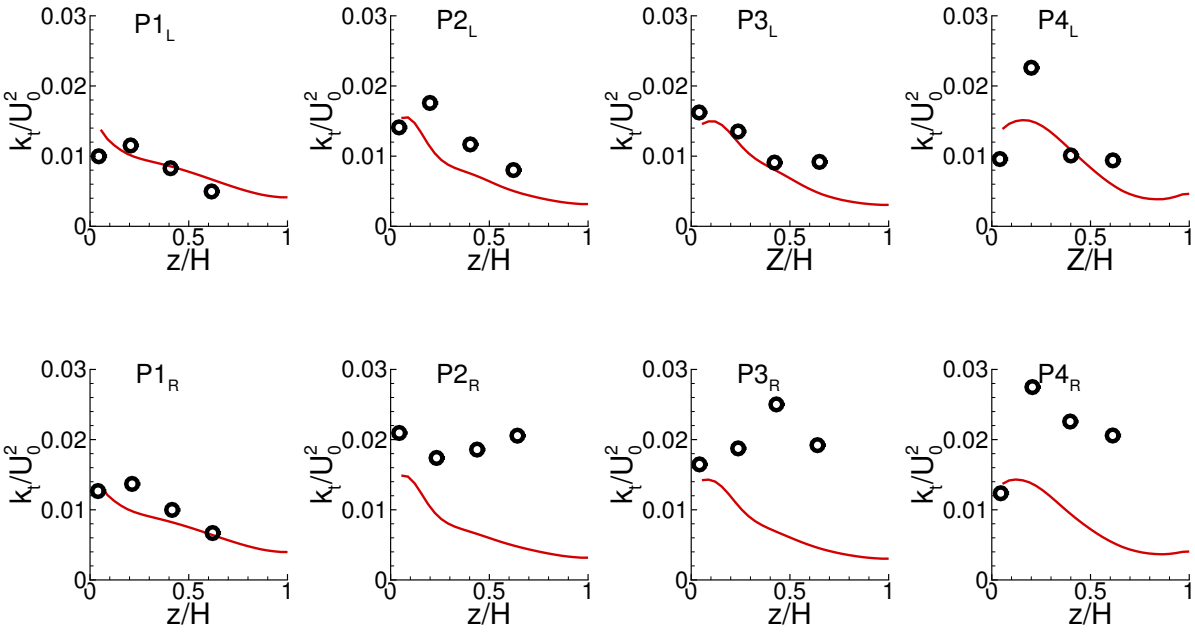


Figure 26: Comparisons of measured (open circles) and calculated turbulence kinetic energy profiles in the depth direction at four stations (see Fig. 14(b) for measurement locations). Profiles symmetrically located on both sides of the foundation are marked with L (left side) and R (right side) subscripts, respectively.

We hypothesize that the striking asymmetry in the turbulence structure is the result of the previously discussed slight misalignment of the approach flow relative to the axis of the piers. Such misalignment could be further exacerbated by the complex pier geometry and drastically change the intensity of the vortex shedding in the left and right shear layers emanating from the obstacles. To explore this hypothesis we have carried out simulations for case C4 in which the approach flow was skewed by 1.8° (this specific flow angle is selected based on the estimated misalignment of the flow in the experimental flume) clockwise relative to the axis of the foundation. An overall view of the time-averaged velocity field for this case is shown in Fig. 27, which depicts axial velocity and turbulence kinetic energy contours at one horizontal plane and clearly shows the thickening of the boundary layer on the right side of the foundation as a result of approach flow skewness. The results for cases C1 and C4 are compared with each other and the experiments in Figs. 28 to 30, which are in exactly the same format as Figs. 24 – 26 above. As seen in Fig. 28, the skewness of

the approach flow has a negligible effect in the velocity field upstream of the foundation. Discrepancies between the results of cases C1 and C4 begin to appear within the piers and are most significant in the wake region (section F6) where the profiles for case C4 exhibit a clear shift to the right (see Fig. 29), a trend which is in broad qualitative agreement with the measurements. The imposed slight misalignment of the upstream flow has a far more dramatic and pronounced effect in the turbulence kinetic energy profiles, which are shown in Fig. 30. Even though significant discrepancies between measurements and simulations still remain, the simulated flowfield for case C4 exhibits qualitatively the main trend observed in the data: the steep rise of turbulence kinetic energy in the right side of the foundation and the gross asymmetry of the turbulence structure. Therefore, the results presented in Figs. 28, 29, and 30 clearly show that even a small misalignment of the approach flow has only a small to moderate effect on the mean velocity field but drastically impacts the turbulence structure. Based on these results we argue that unless the approach flow conditions can be determined with a great degree of certainty in the experiment, reproducing the measured turbulence field around a foundation as complex as that considered herein computationally could be very difficult if not impossible. Recall for example that the measured velocity profiles upstream of the foundation (see Fig. 28) show small deviations from symmetric and uniform flow across the channel depth. These small three-dimensional disturbances, which have not been accounted for in the simulation, could very well be responsible for the discrepancies between measured and predicted turbulence kinetic energy fields.

3.2.3 Flow Patterns and Scour

In this section we seek to establish links between the complex hydrodynamics induced by the bridge foundation as they emerge from our numerical simulations and the scour patterns that result under the same flow conditions in a laboratory experiment with the same foundation mounted on an erodible bed. Since our computations have assumed a fixed, flat bed our discussion herein is only qualitative. It is strictly aimed at underscoring the complexity of the hydrodynamic processes that drive the scouring process in real-life bridge foundations and at providing some guidance for future extensions of the model to

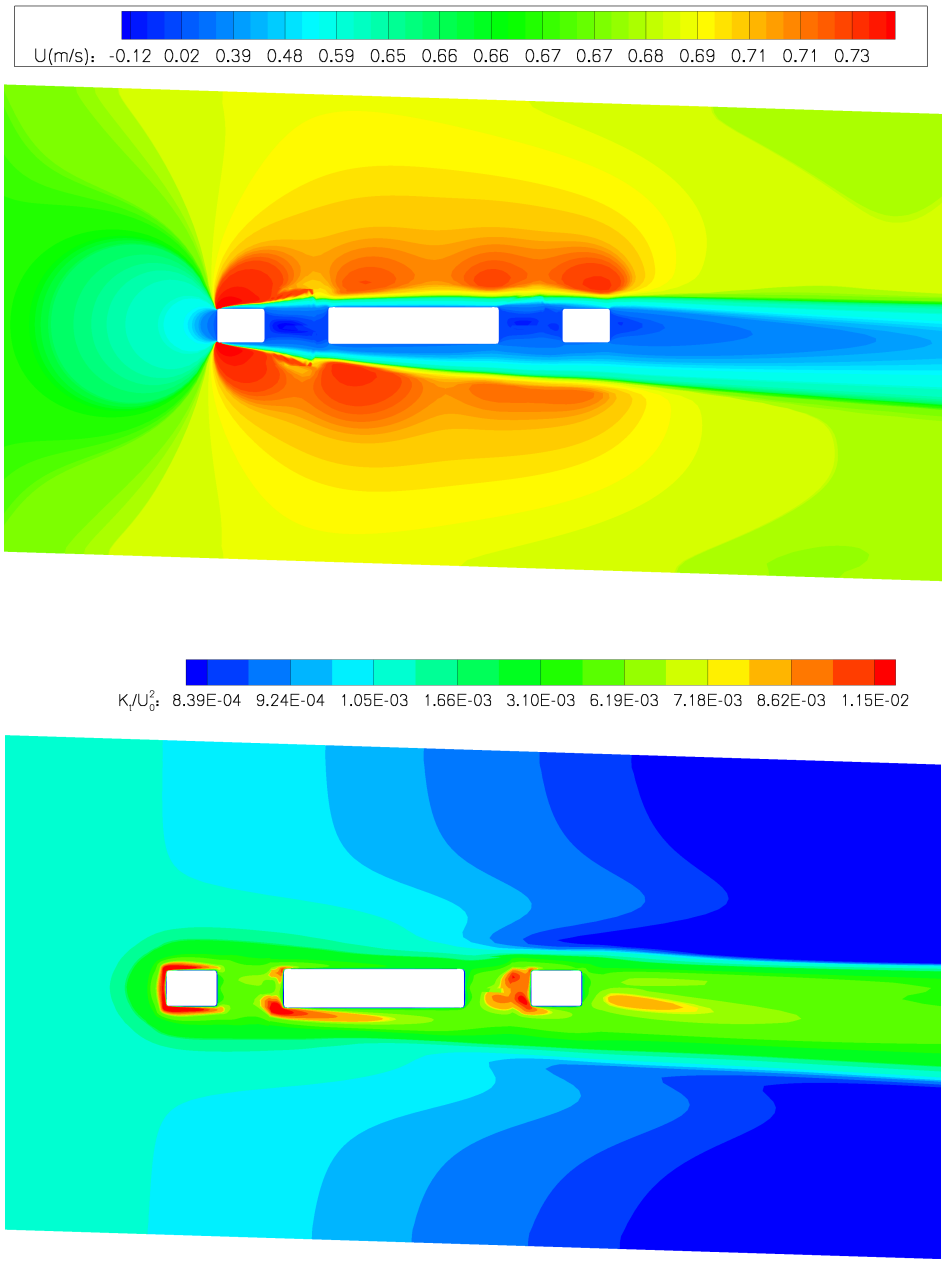
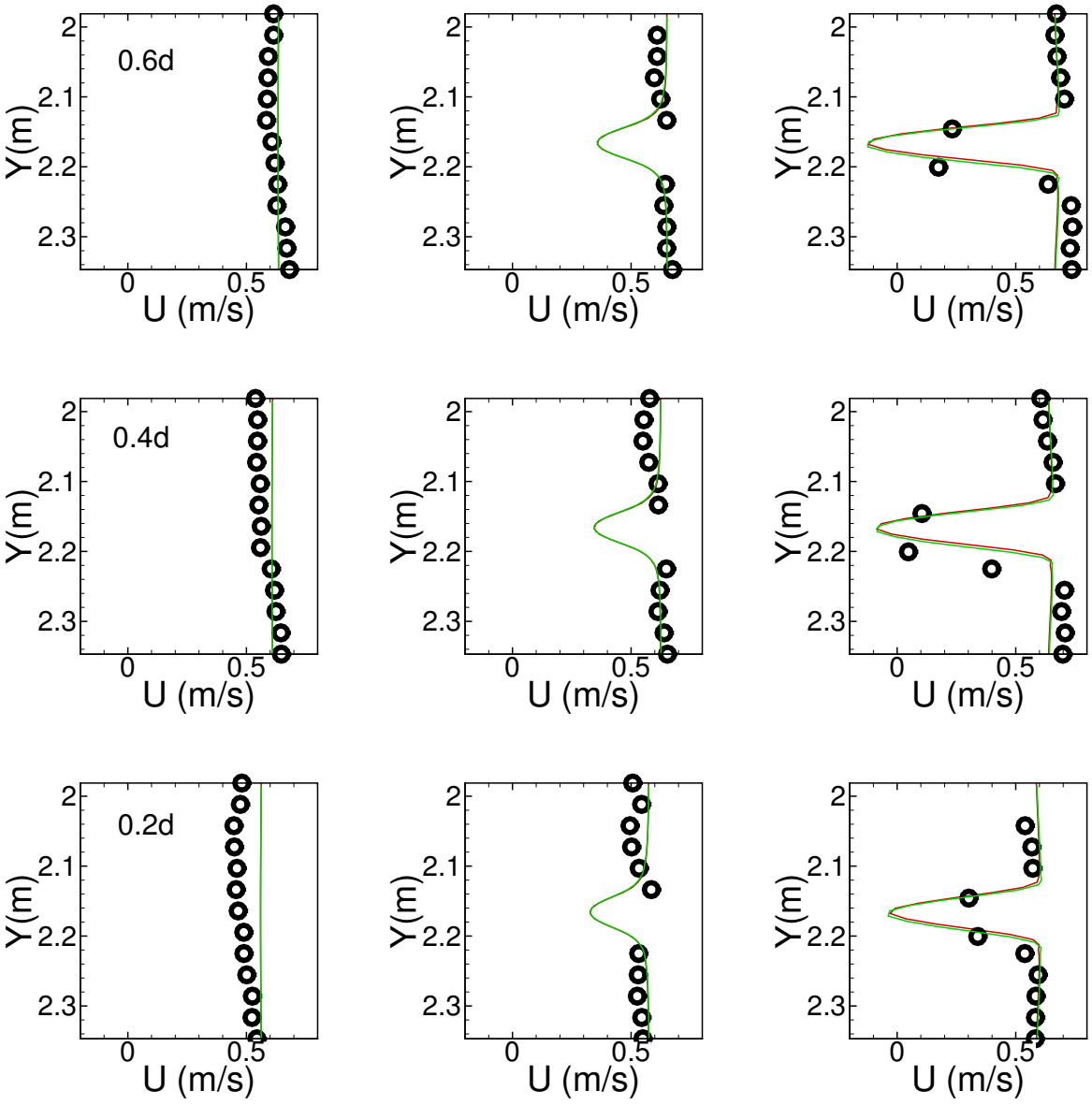
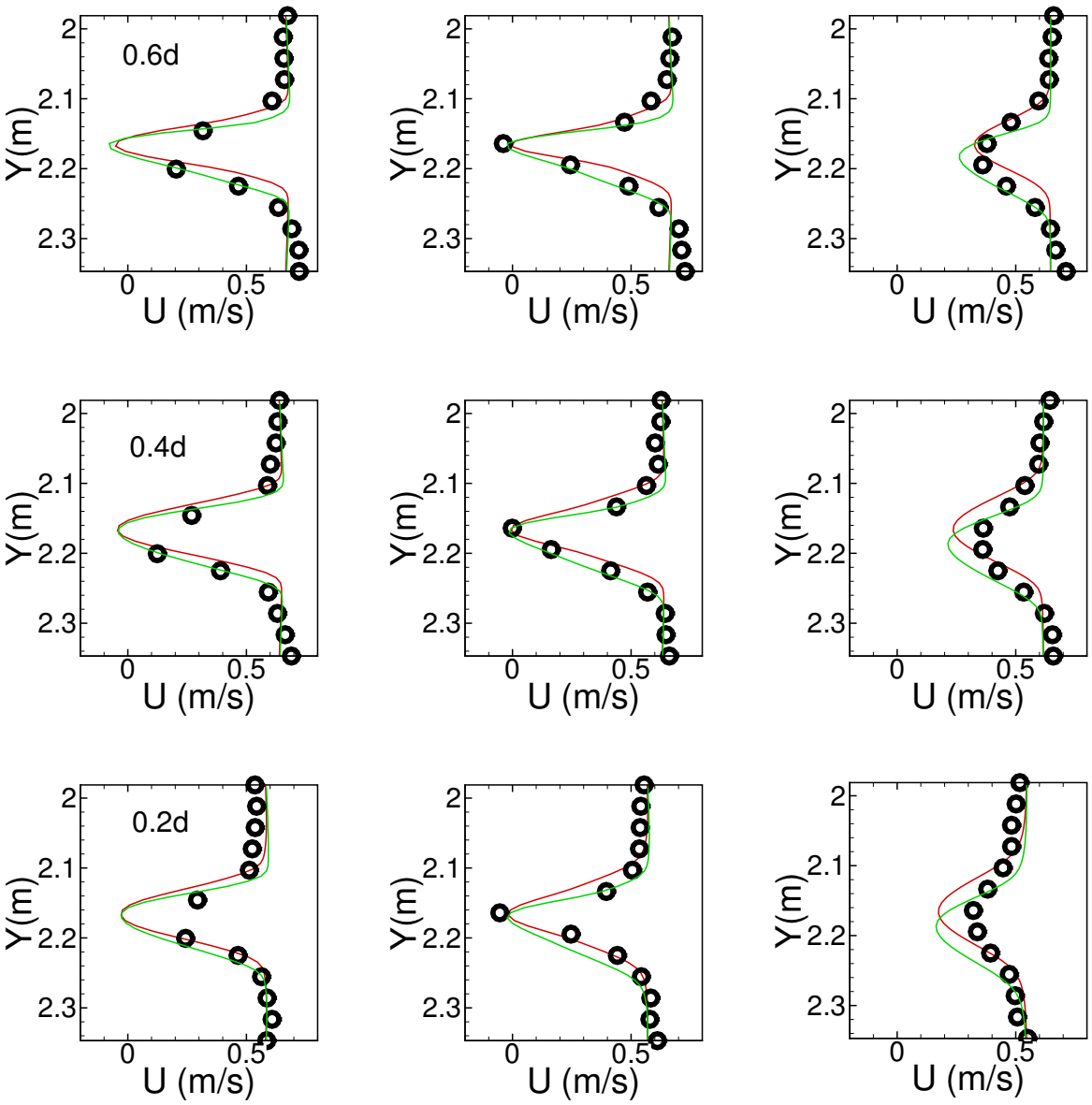


Figure 27: Time-averaged contours of streamwise velocity (top) and total turbulence kinetic energy k_t/U_0^2 (bottom) at $z = 0.7H$ from bottom for case C4 (skewed inflow condition).



(a) F1 (b) F2 (c) F3

Figure 28: Comparisons of measured (open circles) and calculated streamwise mean velocity profiles in the transverse direction at various depths and streamwise locations upstream of the piers for cases C1 (red), and C4 (green). Streamwise locations: a) F1; b) F2; c) F3 (see Fig. 14(b) for measurements locations). Depth locations: From bottom to top, $0.2H$, $0.4H$ and $0.6H$ respectively.



(a) F4 (b) F5 (c) F6

Figure 29: Comparisons of measured (open circles) and calculated streamwise mean velocity profiles in the transverse direction at various depths and streamwise locations upstream of the piers for cases C1 (red), and C4 (green). Streamwise locations: a) F4; b) F5; c) F6 (see Fig. 14(b) for measurements locations). Depth locations: From bottom to top, $0.2H$, $0.4H$ and $0.6H$ respectively.

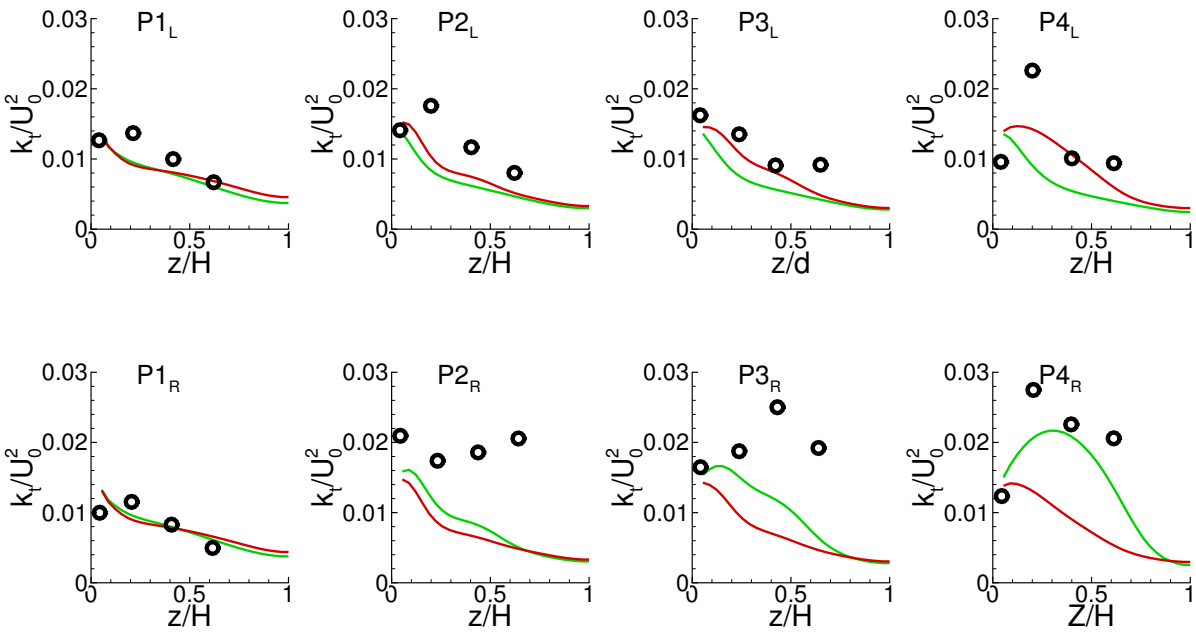


Figure 30: Comparisons of measured (open circles) and calculated turbulence kinetic energy profiles in the depth direction at same locations as in Fig. 20 for cases C1 (red) and C4 (green).

develop a scour-prediction numerical tool.

The equilibrium scour patterns obtained from the laboratory experiments are shown in Fig. 31. As seen in this figure, a scour trench develops that surrounds the entire foundation with the deepest scour occurring upstream of the first pier. Another region of relatively deep scour within this trench is also observed just upstream of the last pier. It is important to note the overall asymmetry of the scour patterns, which becomes more pronounced downstream of the first pier. Such asymmetry is in accordance with the previously discussed impact of approach flow skewness on the structure of the foundation-induced turbulence.

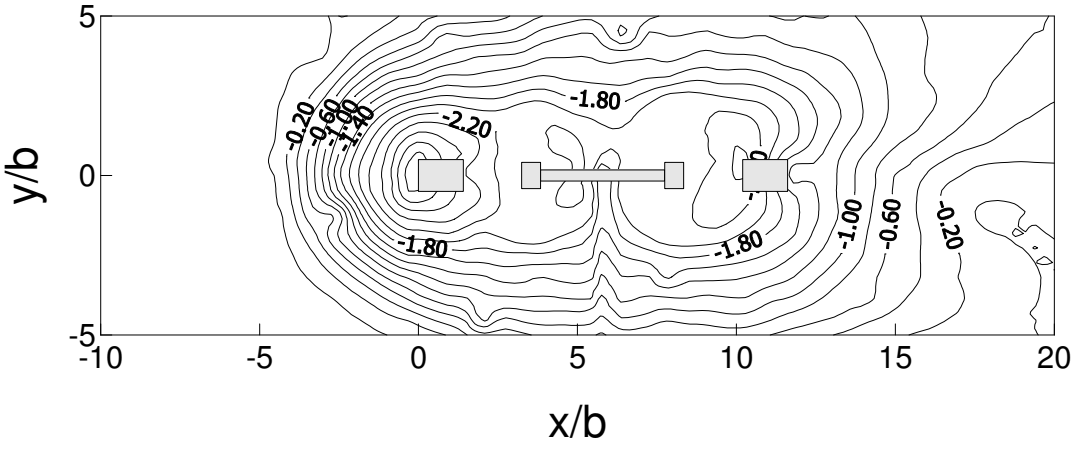


Figure 31: Measured scour contours at equilibrium state (U_0/U_c= 0.94, H/b = 4.16) (Sturm et al., 2004).

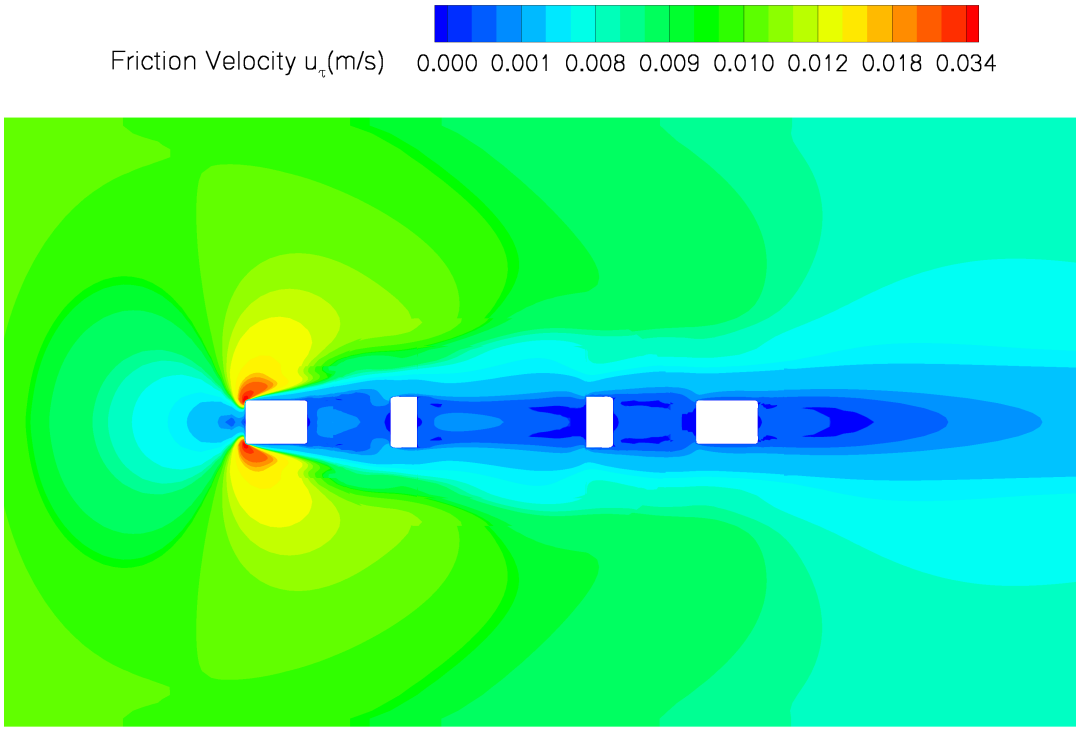


Figure 32: Contours of calculated time-averaged shear velocity (case C1).

Most available sediment transport models employ the concept of critical bed shear stress, the so-called Shields parameter, to define the threshold for incipient sediment grain motion. It would, thus, be instructive to examine the simulated bed shear stress contours for the flat-bed case as that would tend to identify regions in the flow where the scouring process would be initiated. The calculated time-averaged shear velocity contours are shown in Fig. 32. Two pockets of maximum shear velocity are observed at the two upstream corners of the first pier. The calculated shear velocity levels within these pockets are at least one order of magnitude greater than the shear velocity levels within the rest of the foundation. This trend is to be expected since the last three piers are embedded within the wake of the first pier and the flow in their vicinity is, thus, dominated by large-scale, three-dimensional separation and flow reversal. The pockets of large shear velocity correlate well with the region of maximum scour depth surrounding the first pier.

It is evident from Figs. 31 and 32, however, that the distribution of bed shear stress

alone can not account for the complexity of the scour patterns observed in the experiment. To further elucidate the role of foundation-induced hydrodynamics on scour, we plot in Fig. 33 contours of vertical time-averaged velocity at a horizontal plane very close to the channel bottom ($0.01H$). The vertical velocity component is a good indicator of the complexity and three-dimensional structure of the vortical patterns near the foundation. For example, a pocket of negative vertical velocity component near a pier indicates that the flow along the obstacle is directed toward the bed. For continuity to be satisfied, however, such a pocket of downflow must be accompanied by a horizontal flow along the bed directed away from the obstacle, which would tend to sweep bed material away from the obstacle and promote scour. Alternatively, a pocket of positive vertical velocity around a pier suggests a vertical upwelling along the pier away from the bed and must be accompanied by a region of horizontal flow directed toward the obstacle. Such secondary flow patterns would tend to sweep bed material toward the obstacle and lead to local deposition. To better illustrate these flow patterns at the horizontal plane, we show in Fig. 34 the limiting streamlines (or skin-friction lines) corresponding to the vertical velocity contours shown in Fig. 33. As seen in Figs. 33 and 34, the region of negative vertical velocity around the first pier is indeed accompanied with a horizontal flow along the bed directed away from the pier. The topology of the limiting streamlines in this region, which consists of the C-shaped separation line surrounding the obstacle, the saddle node delineating the approach and near obstacle flows, and the half saddle node on the upstream face of the obstacle, is characteristic of the horseshoe vortex system induced by the pier. Similarly, the pocket of positive vertical velocity at the downstream end of the first pier is indeed accompanied by a near-wall flow directed toward the pier, which emanates from the half saddle node on the upstream face of Pier 2. It is also worth noting from Fig. 34 the complexity of the topology of the limiting streamlines around Piers 2, 3, and 4, which is characterized by the presence of pairs of saddle foci in the wake of each pier. These saddle foci tend to sweep flow toward each pier and are thus the footprints on the bed of vertically oriented, tornado-like vortices. Such three-dimensional, vortical structures are seen in the snapshot of instantaneous particle paths shown in Fig. 35, which further underscores and clarifies the complexity of the flow

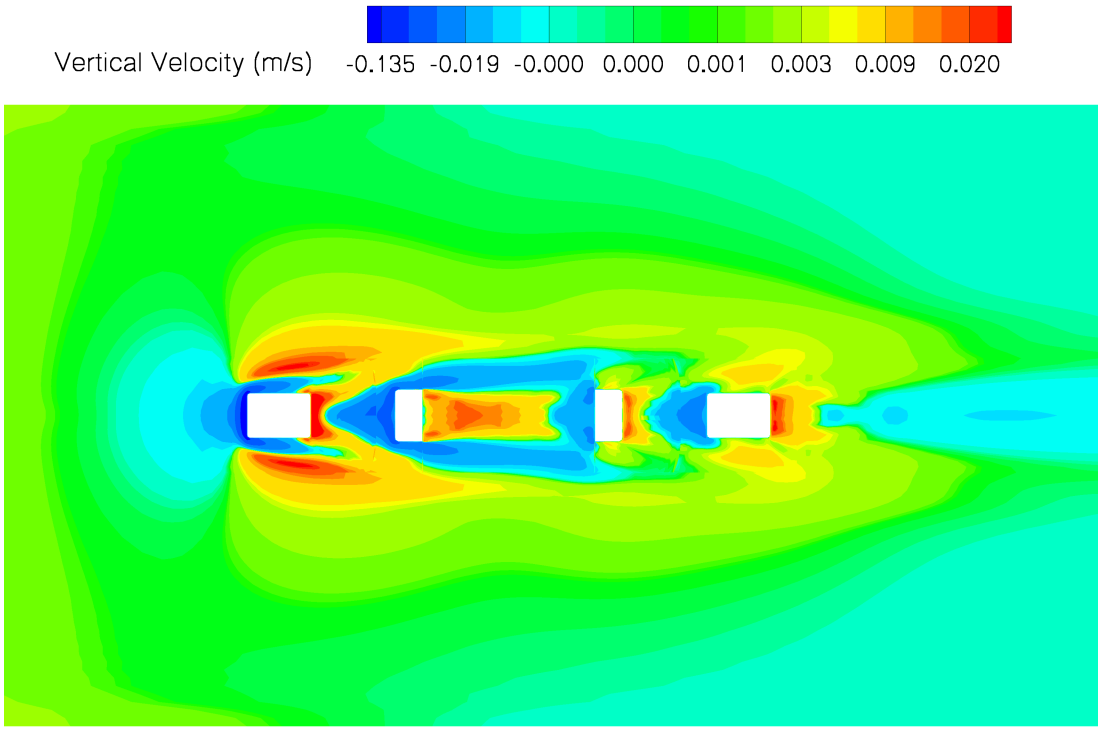


Figure 33: Contours of calculated time-averaged vertical velocity (case C1).

suggested by the two-dimensional plots of vertical velocity and limiting streamlines.

Juxtaposing now the near-bed flow patterns shown in Figs. 32 to 34 with the observed scour map shown in Fig. 31 reveals that the regions of deepest scour at the front of pier 1 correlates well with the pocket of negative vertical velocity, the associated region of near-bed flow away from the pier, and the two pockets of maximum bed shear stress. The second region of deep scour located at the face of pier 4 correlates well with the pocket of negative vertical velocity even though no appreciable levels of shear velocity exist in this region. Another interesting feature of the scour patterns visible in Fig. 31 is the characteristic C-shaped structure of the bed-elevation contours at the downstream end of piers 1 and 4, which reveals the presence of two small ridges of local sediment deposition with less scour adjacent to the ridges. These ridges appear to correlate well with the pockets of positive vertical velocity in the downstream end of piers 1 and 4, thus, supporting our previous qualitative discussion on the role of local hydrodynamics on the sediment transport processes.

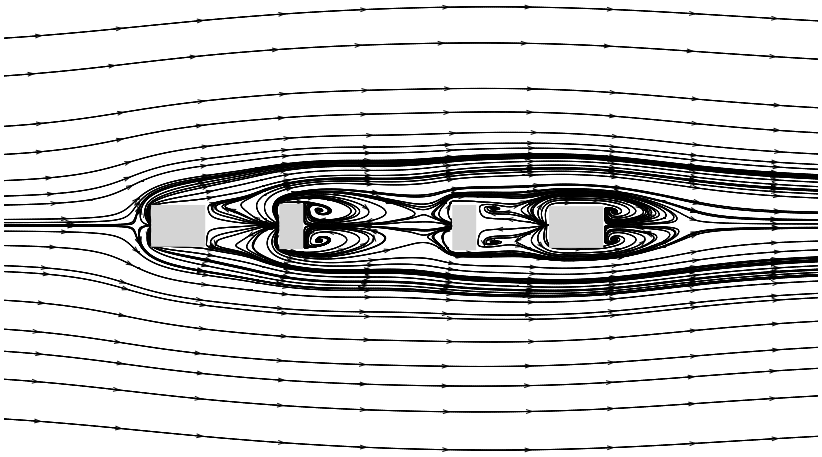


Figure 34: Calculated time-averaged limiting streamlines (case C1).

As we remarked at the start of this section, our discussion herein is only qualitative. The simulated flow patterns for the flat bed case can only provide some indication as to where and how scour will originate. The complex deformation of the channel bed in the vicinity of the foundation, as revealed by the experiments, will undoubtedly alter the local hydrodynamics which will in turn affect the rate of sediment transport and deposition. Our discussion, however, serves to clearly underscore that simplistic sediment transport models relying exclusively on the critical, bed-shear stress concept may not be adequate for modeling scour at real-life bridge foundations.

3.3 Flow past two pier bents on flat bed

The last case to be simulated is turbulent flow past two pier bents placed at the same spanwise distance as in the complete bridge section and mounted on a flat bed. The objective of these computations are to illustrate that our method is capable of obtaining unsteady solutions for very complex, multiple-pier arrangements and is, thus, applicable to model complete bridge sections. The geometrical configuration and spanwise spacing of the two bents is based on the same Chattahoochee River bridge near Cornelia, GA. The geometry is shown in Fig. 36 and is obtained by simply duplicating and translating in the spanwise direction the five blocks around the single pier bent shown in Fig. 15 and patching a square

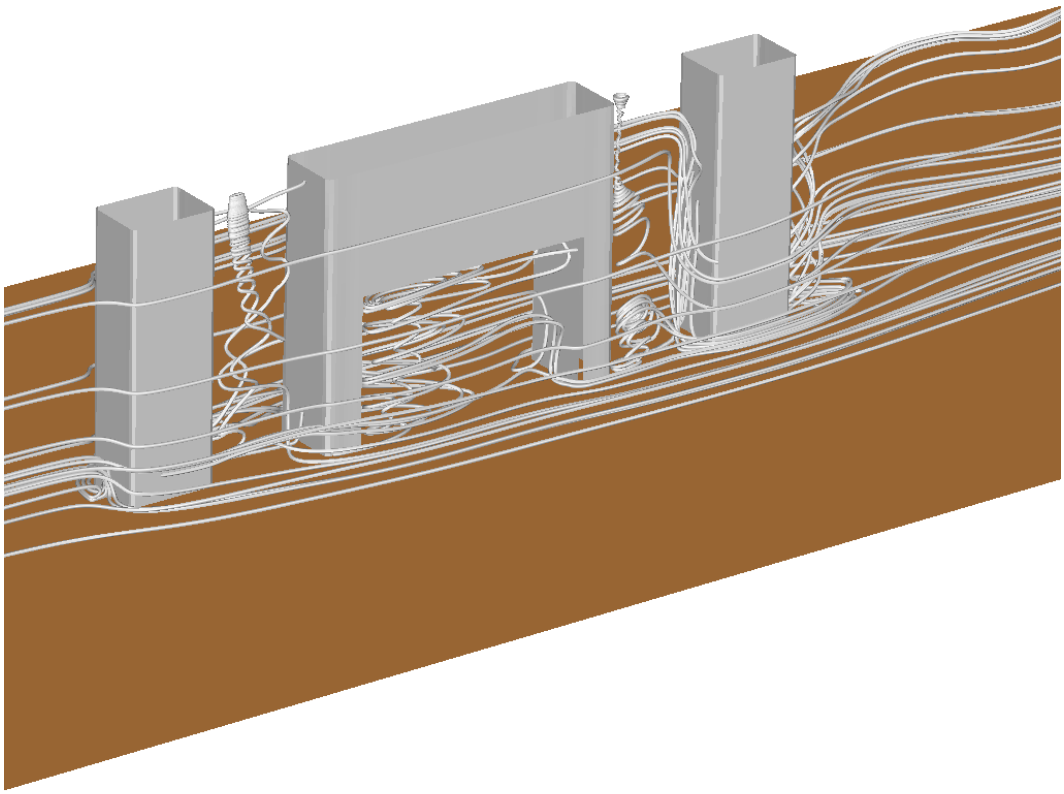


Figure 35: Snapshot of instantaneous (resolved) streamlines depicting the complex web of large scale vortices within the foundation (Case C1).

block in the middle to connect these two structures. The total grid size for this case is 1.9×10^6 grid nodes. Calculations were carried out for Re = 28516 (based on the upstream bulk velocity and the width of the first bridge pier) using the standard k-e model with wall functions to close the URANS equations. A dimensionless time step equal to 0.1T is used for the computations. Only small samples of the computed results are shown below since, as stated before, the main objective of this section is to underscore the capabilities of the numerical method for a very complex configuration. The instantaneous streamwise velocity contours on a horizontal plane just below the water surface are shown in Fig. 37. As seen in this figure, despite the relatively long Spanwise distance between these two bents, there appears to be a weak interaction and coupling between the vortical structures emanating from the solid walls of piers. To quantify the interaction, we calculated turbulent statistics from the simulated flowfields. Fig. 38 shows the turbulence kinetic energy profiles along the depth direction at four locations on both sides of the bottom pier bent (the locations are P1, P2, P3 and P4 shown in Fig. 14). These locations are symmetric about the symmetry axis of the bent. If there were no coupling between these two bents, the turbulence kinetic energy profiles would be symmetric with respect to the streamwise axis of the bent. Yet, and as shown in Fig. 9, there are significant and very pronounced asymmetries between the two sides of the pier bent. In particular, the levels of turbulence kinetic energy on the lower (right) side of the lower bent are considerably more elevated than those on the left side of the bent. This effect appears to be very similar to the previously discussed effect of approach flow skewness on the distribution of turbulence kinetic energy for the single bent case (see Fig 30). To further illustrate this similarity we show in Fig. 39 contours of turbulence kinetic energy at the water surface for the two pier bents case. Based on these results, we can conclude that the presence of the second bent causes a blockage effect and alters the angle at which the flow impinges on the other bent—of course this effect is mutual as both bents experience the same effect due to their coupling of their flows. The resulting small skewness of the approach flow alters the three-dimensional separation patterns causing a more intense shear-layer vortex shedding on one side of the bent, which results in higher levels of turbulence kinetic energy. The apparent extreme sensitivity of

the turbulence statistics on the approach flow characteristics is a very important finding of this work. Recent sediment transport modeling approaches rely increasingly on stochastic models that emphasize the importance of turbulence structure on the prediction method (Papanicolaou et al., 2002). Our findings, therefore, suggest that quantitatively accurate predictions of scour in real-life bridge foundations may require taking into account the large-scale bathymetry of the natural reach and simulate the complete bridge section. Our method, therefore, provides the first powerful computational framework for developing such scour prediction methodology.

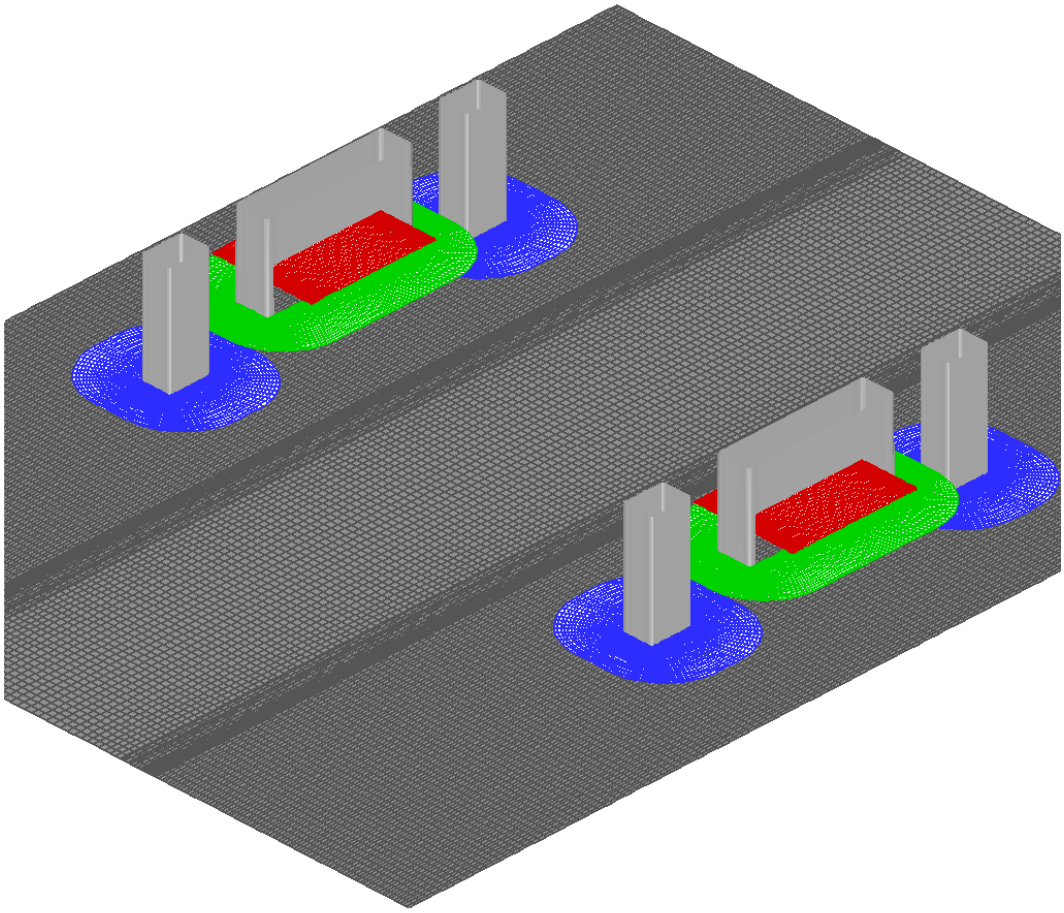


Figure 36: Numerical geometry for two bents of bridge piers flow simulation

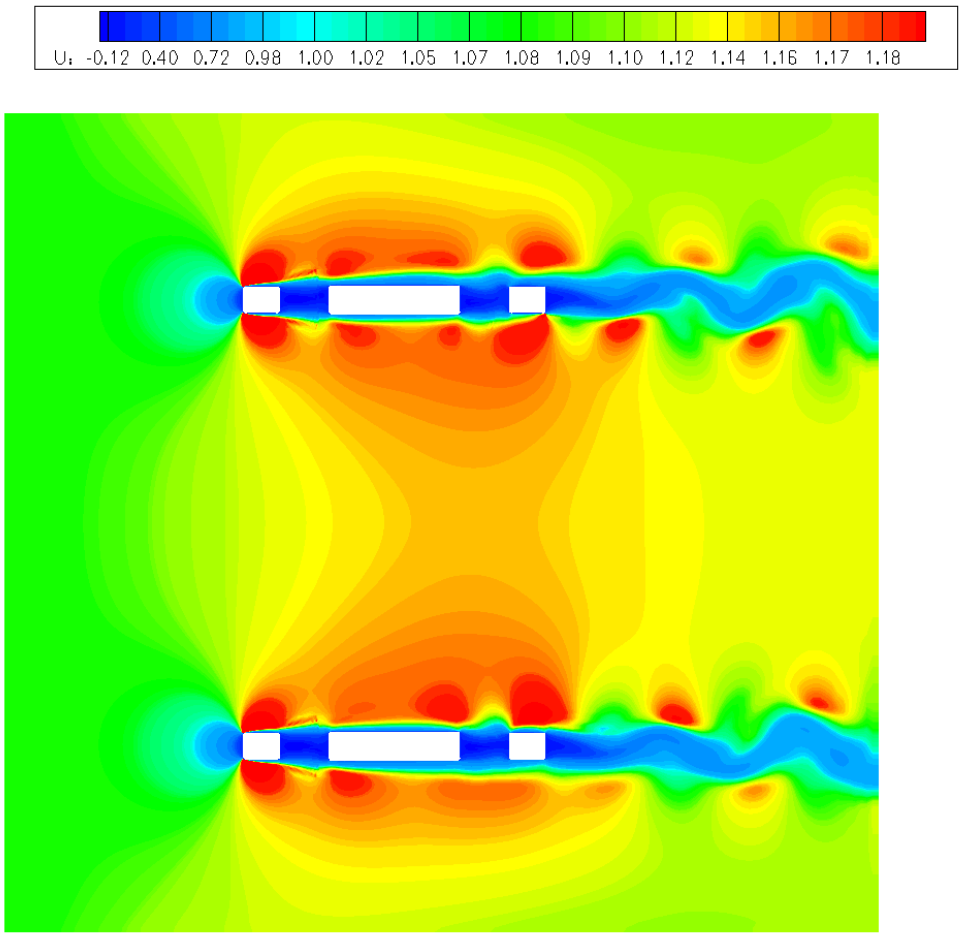


Figure 37: Top view of instantaneous streamwise velocity contours at $y = 0.7H$

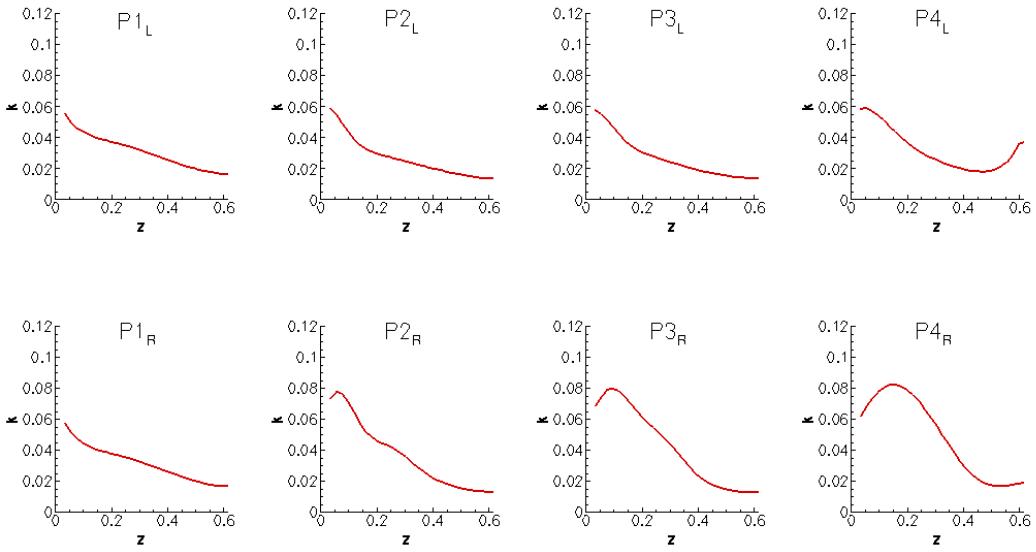


Figure 38: Turbulence kinetic energy profile along the vertical direction (locations are shown in Fig. 14

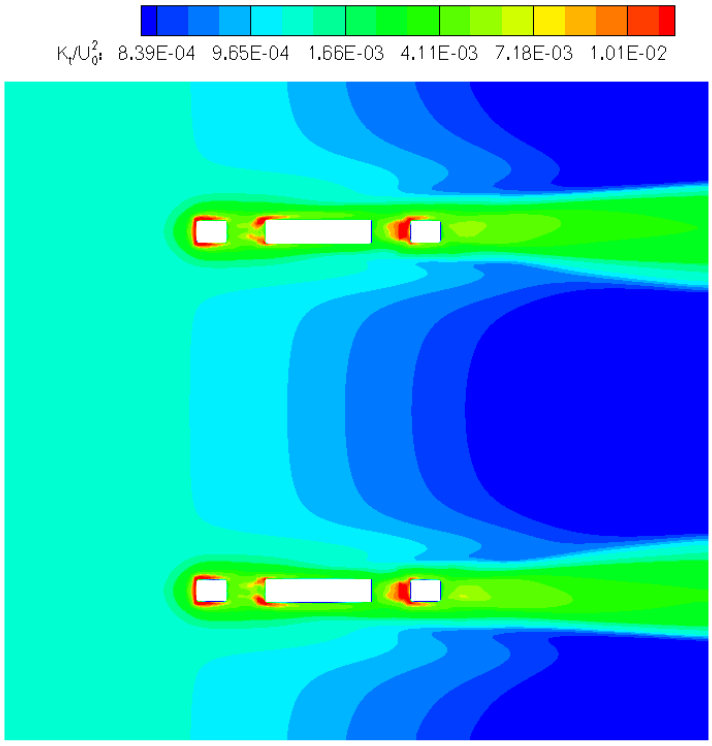


Figure 39: Total turbulence kinetic energy kt/U_0^2 at $z = 0.7H$ from bottom (two bents case)

CHAPTER IV

FLOWS IN MECHANICAL HEART VALVES

4.1 Introduction

The heart is a vital organ. Its major function is to pump the blood to all tissues in our body through the circulation system, thus delivering oxygen and nutrients to the body tissues and washing away the wastes. Figure 40 shows a schematic of a healthy human heart, which has two sides with two chambers at each side. The upper chambers are called atria and the lower two are called ventricles. Connected to the left ventricle is the aorta, the largest artery in the body. It carries oxygen–rich blood from the heart to the body. The vessel connected to the right ventricle is called pulmonary artery, which carries oxygen–poor blood from the right ventricle to the lungs. A normal heart has four heart valves: the tricuspid valve between right atrium and right ventricle; the mitral valve between left atrium and left ventricle; the pulmonary valve between the right ventricle and the pulmonary artery; and the aortic valve between the left ventricle and the aorta. Each valve consists of flaps (cusps or leaflets), which open and close at only one direction. Through the movement of their flaps, the valves regulate the blood flow through the heart and ensure the blood runs only in one direction.

Healthy heart valves open and close fully. However, they may not always work properly because of heart valve disease. The cause of heart valve disease may be linked to one of the following:

- congenital defect (born with an abnormal valve)
- Rheumatic fever
- bacterial infections

Problems associated with heart valve disease include:

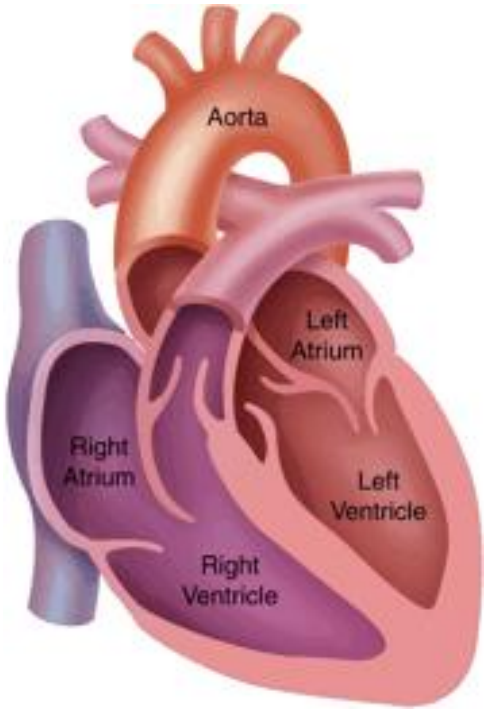


Figure 40: Anatomy of a healthy human heart (from www.nebraskamed.com)

- stenosis: the valve opening becomes narrowed thus inhibiting the ability of the heart to pump blood to the body;
- regurgitation: the valve does not close completely, causing blood to flow backward through the valve;

Medication and surgery may be used to fix defective valves. However, in some cases, these remedies may not be sufficient for fixing the defective valves and the native valve has to be replaced with a prosthetic valve in order for the patient to have a normal life. Currently, there are two types of prosthetic valves used for dysfunctional native valve replacement: mechanical valves, which are made entirely from biocompatible synthetic materials; or tissue (or bioprosthesis) valves, which are made from specially processed and shaped animal tissue. The major advantage of mechanical valves is their durability. Because of the usage of the synthetic materials, the mechanical properties of these valves will not degrade. These valves usually last through out the patient's lifetime and does not require replacements. However, due to the man-made and thus non-physiologic nature of these valves (both the structure and

materials), they are prone to the development of blood clots. Recipients of mechanical valves have to take anticoagulation medication (blood thinner) to prevent thromboembolism and these anticoagulation medications increase the risk of bleeding in the body. On the other hand, tissue valves, because of their origins, closely resemble the body's natural valve. Therefore, lifelong anticoagulation therapy is not required and patients can have a better life quality. However, the mechanical properties of these valves degrade rapidly and they are prone to calcification that we see as heart valve disease develops (Black and Drury, 1994). Tissue valves may need to be replaced every 10 to 15 years. For this reason, the implantation of tissue valves are often limited to elder patients. Mechanical valves are more widely implanted for their long durability.

The first mechanical heart valve as a prosthesis for dysfunctional heart valve was implanted in 1952. Since then, many different designs of MHVs have been developed. Currently the most popular design is the bileaflet valve, which has better hemodynamics performance comparing with earlier generation designs. However, implantation of all existing MHVs could lead to major complications, including thromboembolism, hemolysis and tissue overgrowth. Although the exact mechanisms leading to these complications are yet not fully understood, they are believed to be strongly related to the non-physiological local flow patterns and the elevated turbulence shear stresses induced by the prosthesis. All MHVs use an occluder, which could be a caged–ball, a tilting disc or two semi–circular leaflets, to mimic the open/close function of the natural valve. The occluder partially blocks the flow channel, leads to regions of stagnation and flow separation in the wake and enhances turbulence. The elevated turbulence shear stresses could cause lethal or sublethal damage to blood cells, such as rupture of red blood cells and platelet activation, and initiate the formation of thrombi. The local slow flow regions enhance the aggregation of platelets, thus, further facilitating the growth of thrombi. Therefore, to improve the design of MHV it is essential to investigate and understand the complex hemodynamics they induce. The following section provides a review of previous experimental and numerical research on MHV related problems.

4.1.1 Experimental Work

A number of experiments have been conducted to study the MHV related flow problems. These experiments can be divided into two categories: experiments focused on establishing links between the clinical complications and the hemodynamics; and experiments aimed at investigating the complex hemodynamics induced by the valves. Experiments in the former category have shown that exposure to high shear stress can cause lethal or sublethal damages to blood cells, including red blood cells and platelet. Smith et al. (1972) and Stein and Sabbah (1974) reported that high turbulence shear stress can cause blood cell and platelet damage, which eventually lead to thrombus formation. Sutera et al. (1972) investigated the response of red blood cells to shear stress produced within a concentric cylinder viscometer. They found the critical shear stress to cause hemolysis by long time exposure was approximately 1500 $dyne/cm^2$. Sallam and Hwang (1984) showed that in a submerged turbulent jet lethal red cell damage could result from turbulent shear stresses of around 4,000 $dyne/cm^2$ for an exposure time of less than six seconds. With the existence of foreign surfaces, damages of red blood cells have been found at a much lower threshold of shear stress (on the order of 10 $dyne/cm^2$ (Mohands et al., 1974)). The critical shear stress which can cause platelet damage is substantially lower than that of the red blood cells. In 1975 Brown et al. (1975) performed experiments to test a relationship between thrombus formation and a known shear stress. They found that platelets exposed to shear stresses in excess of 50 $dyne/cm^2$ release a significantly greater amount of the agonist ADP and tend to aggregate faster than platelets exposed to lower shear stresses. In 1985, Wurzinger et al. (1985) studied platelet responses to shear stresses acting over a time scale of milliseconds, a more appropriate physiologic exposure time than previous investigators had studied. Using a rotating Couette viscometer they subjected platelet-rich human plasma to a range of shear stresses for a duration of 7 ms to 700 ms. By measuring levels of the protein b-thromboglobulin, the presence of which is considered indicative of platelet activation, they found that a laminar shear stress as low as 1,000 $dyne/cm^2$ acting for 110 ms caused significant liberation of this protein.

Yoganathan et al. (1986) carried out the first experiments in the second category by employing 2D LDV to investigate the pulsatile forward flow fields of the SJM bileaflet valve in the aortic position and the Medtronic-Hall and Bjork-Shiley tilting disc valves. They reported maximum turbulent shear stresses downstream of the leaflets ranging from 1,200 $dynes/cm^2$ for the SJM valve to 2,000 $dynes/cm^2$ for the Medtronic-Hall valve. Using 1D LDV, Chandran et al. (1983) investigated the flow past different caged ball and tilting disc aortic valve prostheses. The study showed that the velocity profiles and turbulent shear stress magnitudes downstream from the tilting disc prostheses are dependent upon the opening of the disc and that the turbulent normal stresses downstream from the ball caged prostheses were smaller than those behind tilting disc valves. Furthermore, comparison of measurements with steady flow at flow rates comparable to peak pulsatile flow rate show that the turbulent normal stresses are larger by a factor of two in pulsatile flow with a frequency of 1.2Hz. Chandran et al. (1985) also investigated the flow downstream of the Bjork-Shirley tilting disc valves in a human aorta model illustrating the influence of valve orientation on the downstream velocity profile in the mid-arch and brachio-cephalic arterial branch of the aorta. The effect of valve orientation on the hemodynamics of the SJM and Duromedic (DM) valves was studied by Fatemi and Chandran (1989) using 1D LDV. Differences in the velocity profiles of the two valves were attributed to variation in geometry, as the DM valve had thicker leaflets, smaller angle of leaflets opening and leaflet curvature. Schoephoerster and Chandran (1991) investigated the flow dynamics in various caged ball, tilting disc, bi-leaflet and polyurethane mitral valves using LDV. The turbulent shear stresses magnitudes did not exceed the reported value for hemolytic damage, however the shear stresses were of the same order of magnitude as the valves reported in the literature for the onset of platelet lysis. Walker and Yoganathan (1992) looked at the Omni-Carbon tilting disc valve and the Duromedics bileaflet valve. The Omni-Carbon valve design produced turbulent shear stresses up to 2,000 $dynes/cm^2$. The velocity profile taken across the central orifice of the Duromedics bileaflet valve showed a large region of flow separation around its pivots. Turbulent shear stresses as high as 1,700 $dynes/cm^2$ were found adjacent to these separated zones. Fontaine et al. (1996) were the first to study the Bjork-Shiley

and SJM aortic valve flow fields with a 3D LDV. They reported relatively small differences between Reynolds stresses calculated by three-component vs. two-component LDV-within 10 to 20 percent. They also found that the Reynolds shear stresses calculated along the velocity measurement axes can underestimate the principle Reynolds shear stress by as much as 100%.

The investigation of the effects of valve closure flow on the blood began in 1989, when clinical failures were reported for a small number of Edwards-Duromedics valves (Klepetko and Moritz, 1989). Pitting and erosion, markers of cavitation damage, were noted on these valves. Although cavitation has not been suggested as a cause of failure for any other valve design, examinations of several other explanted valves has revealed pitting of the same nature (Kafesjian et al., 1994; Wieting et al., 1990). Shortly after the failures of the Edwards-Duromedics valves were reported to the scientific community, Graf et al. (1992) performed a series of experiments in an attempt to determine cavitation thresholds for various mechanical valves. These researchers simulated flows and pressures within the circulation for resting and exercise conditions in an in vitro left heart simulator system. Under resting conditions (5 L/min cardiac output and ventricular loading rate of 1500 mm Hg/s), they saw no cavitation proximal to any of the valves they studied. However, when they increased the ventricular loading rate to 3000 mm Hg/s, they observed cavitation just proximal to some valves. Chandran et al. (1994) and Lee et al. (1994) defined a ventricular pressure rise rate during valve closure as a parameter to be defined in experiments for assessing the potential for mechanical valves to cavitation studies. Chandran et al. (1998) measured the large negative pressure transients in vivo during valve closure with mechanical valve mounted in the mitral position in an animal model, similar to those measured in vitro.

The design of mechanical heart valves deliberately includes some degree of leakage, or retrograde, flow upon valve closure. This reverse flow is intended to scour critical areas of the valve, such as the hinges and the areas between the leaflet edges and the housing. Yoganathan et al. (1986) found that under aortic pulsatile flow conditions the peak reverse velocity through the Medtronic-Hall tilting disc was 0.28 m/s, with a peak turbulent shear stress of 680 $dynes/cm^2$. They also found a peak reverse velocity of 0.22 m/s through the

Bjork-Shiley tilting disc with a peak turbulent shear stress of 430 $dynes/cm^2$. These values were measured 10 mm upstream of the valves. They also reported leakage flow 10 mm upstream of the SJM valve and noted a maximum reverse velocity of 0.16 m/s and turbulent shear stresses as high as 325 $dynes/cm^2$. Baldwin et al. (1991) conducted two-component LDV measurements at 100 near-wall positions upstream of the minor orifices of the mitral and aortic Bjork-Shiley Delrin tilting disc prosthetic valves in the Penn State Electric Left Ventricular Assist Device (PSLVAD). The highest turbulent shear stress near the aortic valve was found to be 9,900 $dynes/cm^2$ and the corresponding peak velocity in the leakage jet was 2.8 m/s. Near the mitral valve, the highest turbulent shear stress was 9,000 $dynes/cm^2$ with a peak leakage velocity of 4.4 m/s. These peak turbulent shear stresses and velocities were detected 1 mm upstream of the valves and approximately 0.6 mm from the adjacent wall of the PSLVAD. Lamson et al. (1993) expanded on the Baldwin study by showing that the hemolytic effects of backflow through the PSLVAD contributed significantly to overall hemolysis in the mock circulatory loop through elevated plasma-free hemoglobin levels. Meyer et al. (1997) acquired 3D LDV measurements in the regurgitant flow region proximal of a Bjork-Shiley monostrut valve in the mitral position. The study recorded a maximum velocity of 3.7 m/s and peak turbulent shear stresses of 10,000 $dyne/cm^2$. Subsequent experiments (Ellis, 1999; Ellis et al., 1996b,a, 2000; Ellis and Yogananthan, 2000; Leo et al., 2002) with CarboMedics (CM), Medtronic Hall, and SJM bi-leaflet valves found maximum regurgitant flow velocities ranging from 0.7 to 2.6 m/s, with corresponding maximum Reynolds shear stress of between 450 and 3,600 $dyne/cm^2$. These studies demonstrate that turbulent jets in leakage flow can generate elevated levels of turbulent shear stress, even in bileaflet valves. Recent studies (Ellis, 1999; Ellis et al., 1996b,a; Leo et al., 2002) have shown the geometry of the hinge region in a bileaflet MHV is important in determining the leakage flow properties of the valve. Driven by the pressure gradient across the bileaflet valve during closure, the leakage flow through the constricted hinge region can reach high velocities and be subjected to high levels of turbulent shear stress. These studies suggest that the hinge mechanism is a critical part of the bileaflet MHV since it not only directly influences valve durability and functionality, but the high levels of turbulent shear stress in

the hinge region may also lead to thrombus formation.

The basic geometry of the hinge regions in the MP, SJM and CM are characterized by projections on the leaflets which mate to a similarly shaped recess in the valve housing. The geometry of the MP valve hinge recess has sudden expansion and contraction zones and the flow is unsteady with the formation of vortex structures and regions of flow stagnation (Ellis et al., 1996b). Turbulent shear stresses recorded near the hinge wall in the inflow channel were of magnitude greater than 6000 $dynes/cm^2$, which is an order of a magnitude greater than the level required to damage blood elements (RBC) and lead to the activation of thrombosis (Ellis et al., 1996b). Both the SJM and CM valves have similar semicircular projections on the leaflet, except that in the former the hinge is streamlined and has a curved profile while in the latter the hinge geometry has sharper corners and is less streamlined edges. LDV studies by Ellis (1999) found peak leakage velocities and turbulent shear stresses in a 25 mm SJM standard valve of 3.5 m/s and 7,200 $dynes/cm^2$, respectively. An improved equivalent valve model SJM 23mm Regent achieved lower maximum leakage velocity and turbulent shear stress levels of 1.5 m/s and 2,600 $dynes/cm^2$, respectively (Ellis and Yoganathan, 2000). The highest peak leakage velocity and turbulent shear stress observed in the hinge during the leakage phase in the CarboMedics CPHV models were 3.17 m/s and 5510 $dynes/cm^2$, respectively (Leo et al., 2002).

4.1.2 Computational Work

The application of CFD to MHV flows began in the early 1970s with the 2D immersed boundary method of (Peskin, 1972). The study showed the great potential of CFD as a tool for revealing the complex hemodynamics through heart valves. Since this early work, a number of numerical investigations have been devoted to the flow through heart valves. Huang et al. (1994) performed 2D, time-accurate simulations of a tilting-disk heart valve using a grid resolution of over 90,000 nodes with steady inflow conditions for Reynolds number between 10 and 1000. For flows at Reynolds number greater than 200, periodic vortex shedding from the edge of the valve disk was observed and the intensity of these vortices increases with the rising of the flow Reynolds numbers. Shim and Chang (1997)

performed a three-dimensional analysis of a tilting disk valve at $Re = 210$ and $Re = 450$ under steady flow conditions. They assumed the flow to be symmetric with respect to the vertical plane of symmetry and discretized one-half of the valve geometry using a grid with approximately 32,000 nodes. Their results suggest the presence of longitudinal vortices downstream of the leaflet, whose origin they attributed to pressure gradients in the flow. King et al. (1994) performed steady, 2D, finite-element, laminar simulations of a bileaflet valve for $Re = 3000$. In their 2D geometry, they assumed that the flow was symmetric between the two leaflets. King et al. (1996, 1997) later extended this work to 3D and performed laminar simulations using one-quarter of the valve geometry (i.e., assuming that the flow is symmetric with respect to both symmetry planes of the valve). These simulations used a sinusoidally varying inlet velocity to mimic the acceleration phase of systole with $Re = 3000$ at peak systole. Consistently, most of the previous reported numerical works have either assumed 2D flow or conducted 3D simulations with relatively limited spatial resolutions. Many of the reported 3D simulations have further assumed, presumably for computational expedience, the flow to be symmetric with respect to one or more of the geometrical symmetric planes of the valve. In a recent published work, Ge et al. (2003) challenged these computational expedient assumptions with fully 3-D high resolution numerical simulations of flow through a typical bileaflet heart valve under steady inflow conditions. This work showed that the flow symmetry with respect to the geometrical planes of symmetry of the valve observed at low Reynolds number breaks above a threshold Reynolds number. Upon break of symmetry, unsteady vortex shedding in the wake of the leaflets is observed and complex, three-dimensional vortical structures emerge in the flow. This study suggested that flow through MHVs is far more complex than previously believed. It further illustrated that in–depth understanding of the hemodynamics through MHVs necessitates full 3-D high resolution numerical simulations along with careful numerical validation with experimental measurements.

Despite of the importance of turbulence in the developing of clinical complications, numerical simulations of turbulent flow in heart valves is very limited, primarily due to the high computational resource requirement of such tasks. Stevenson et al. (1985) and

Stevenson and Yoganathan (1985) conducted 2D simulations, with $k-\varepsilon$ model to account for the turbulence effects. Bluestein et al. (2002) investigated the turbulent pulsatile flow past a St. Jude Medical bileaflet heart valve using the FIDAP CFD package. A 2D unstructured mesh consisted of 53,564 computational nodes is used to discretize the geometry around a tilt implanted valve. The turbulent flow effects were modeled with $k-\omega$ model. Vortex shedding and high turbulent shear stress were found in the vicinity and the wake of the valve. Considering the valve's strongly three–dimensional nature of the geometry, such 2D simulations can provide very limited insights into the flow physics and full 3D investigation is required.

4.1.3 The Present Contribution

One of the most prominent flow features of cardiovascular flow is its pulsatile nature. Blood in the cardiovascular system is pumped by the contraction of heart chambers and thus one heart beat comprises a cardiac cycle. The flow velocity in the aorta changes in time and peaks at a value of around 1.35m/s. The typical flow Reynolds number in a healthy human body ranges between 0-7000, based on the aorta diameter and bulk blood velocity. Under normal conditions, blood flows remain laminar through out the cardiac cycle. However, with the existence of mechanical heart valve, due to the unphysiological nature of the valve structure, transition to turbulence is anticipated near peak systole. To investigate, therefore, the MHV flow under physiological condition requires sophisticated numerical models capable of simulating transition to turbulence and re-laminarization in domains with moving boundaries (valve leaflets). Yet, it follows from the above literature review that numerical methods capable of quantitative accurate predictions of physiological MHV flows do not exist today. In fact, to the best of our knowledge no numerical method has convincingly demonstrated up until today that it can accurately predict such flows even under steady state conditions at Reynolds numbers within the physiological range. In this chapter we employ the numerical method developed in this work to study the flow through a St. Jude Medical bileaflet mechanical heart valve with the leaflets fixed at the fully open position over a broad range of Reynolds numbers, $300 < Re < 6000$. We

seek to: 1) validate the numerical method and demonstrate that it can reproduce MHV flowfields in good quantitative agreement with experimental measurements; and 2) provide the first comprehensive insights into the complex hemodynamics of MHV at near peak-systole Reynolds numbers. Even though pulsatile flow effects and motion of the leaflets are not considered herein, our work constitutes an important step toward the development of a numerical simulation framework for MHV flows. Our simulations will lead to a validated and reliable flow solver, which with further work can be used to tackle the next level of complexity: that of pulsatile flow and moving leaflets. Our results will also guide the design of future experiments for obtaining comprehensive experimental data for meaningful CFD validation under physiological conditions. As shown in Fung (1984), in large blood vessels, the shear rate is expected to be high and under such high shear rate, the blood viscosity coefficient asymptotically approaches a constant value. Therefore, in the current study, the flow is assumed to be Newtonian.

4.2 Numerical simulation of MHV flow with fixed leaflets

We carry out a series of numerical simulations for Reynolds numbers both in the laminar and turbulent flow regimes. For the former cases we conduct direct numerical simulations while for the later case we carry out unsteady RANS and DES. In the laminar regime the Reynolds numbers ranges between 300-1250 while the turbulent flow simulations are carried out for Re = 6000, which is near peak flow condition. For all simulations we employ the same valve model and the overset grid methodology developed in this work. In the subsequent sections we first present the results of our laminar simulations followed by the simulations for the Re = 6000 case. For both cases, we discuss the flow physics and compare out computations with laboratory measurements. It is worthwhile to note that in the following discussion, unless otherwise explicitly stressed, all flow variables presented are nondimensionalized. The velocity components are non–dimensionalized with the incoming bulk velocity, U_0, and the time scale by the time scale D/U, where D is the diameter of incoming pipe.

4.2.1 Laminar flow regime

We use a geometric model of a bileaflet MHV as a prosthetic replacement for the aortic heart valve as shown in Figure 41. The primary simplification made in our geometric model is that the aortic root contains a single axisymmetric sinus and not three aortic sinuses. The leaflets in our model MHV closely resemble the St. Jude Medical (SJM) Standard valve, with the hinge mechanism neglected. In the fully open position, the leaflets in the SJM standard valve make an angle of 85 degrees with x-y plane as shown in Figure 41(b,c). Structured overset grids are used to simulate the complex geometrical configuration of MHV. A cross-sectional view of the computational grid is shown Figure 1(c). The aorta geometry is discretized with an O-type, cylindrical grid along with an embedded rectangular Cartesian block to resolve the singular point problem at the center. The dimensions of these two blocks are $67 \times 102 \times 135$ and $21 \times 21 \times 135$ in the i, j, and k directions of curvilinear grid system respectively. The grid for each of the leaflets is a body-fitted H-H grid containing $81 \times 21 \times 65$ nodes (approximately a total of 1.1×10^6 active nodes).

We have carried out direct numerical simulations, by solving the time-accurate Navier-Stokes equations, for various Reynolds numbers in the range 300 and 1250—the Reynolds number is based on the bulk velocity at the inlet and the diameter of the model aorta. It is important to clarify that for all simulated Re no *a priory* assumption was made about the steadiness of the flow. If a steady-state solution indeed existed for a given Re it was reached via full, unsteady simulations. By adopting this approach we sought to identify the threshold Re at which unsteadiness naturally develops and elucidate the mechanisms that excite unsteady modes in the flow with increasing Re. For all simulations we employed a dimensionless time step $\Delta t = 0.01T$ and typically $15 \sim 20$ dual-time iterations were required to reduce the residuals by 3 orders.

For Reynolds numbers less than approximately 325 the time accurate simulations converged to a steady-state after an initial transient status and steady solutions were obtained. This is illustrated in Fig. 42, which shows the time history of transverse velocity component at a point located in the center point of a wake plane. Note that for steady flow with quadrant symmetry, this velocity component at the center of the cross-section should

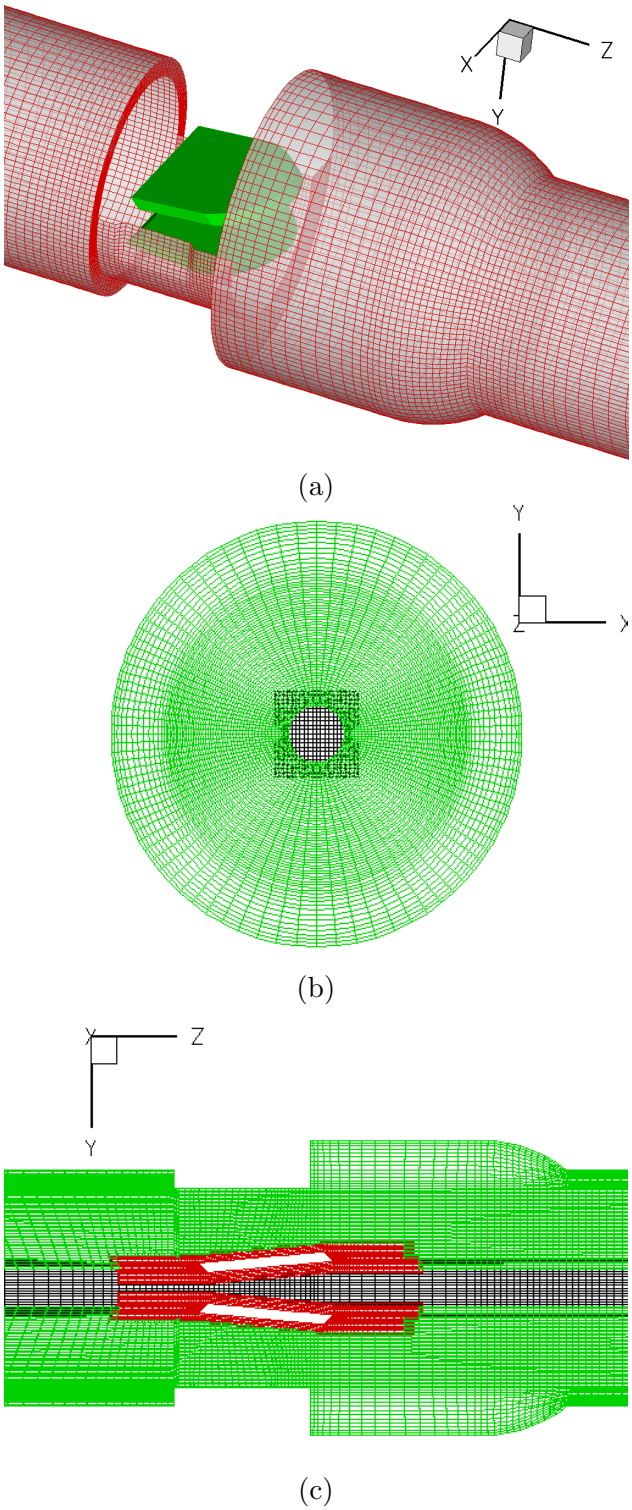


Figure 41: Numerical geometry of the mechanical heart valve: (a) overview of the three-dimensional geometry and grid on aorta wall; (b) cross sectional view shows the overset grid used to avoid singular point in the aorta center; (c) plan view shows the grid used to discretize the vicinity of leaflets

exactly zero and as seen in the figure the zero value is indeed approached at steady state. The flow patterns of the converged steady solution are shown in Fig. 43 and Fig. 49, which depict streamwise velocity and pressure contours at the vertical ($x = 0$) plane of symmetry and streamwise velocity and cross-flow vectors at a cross-section downstream of the leaflets, respectively (see Fig. 41 for axes definition). Fig. 43a shows the streamwise pressure drop through the valve with pockets of high and low pressure appearing at the leading and trailing edges of the leaflets, respectively. The pressure on the leaflets sides facing the valve housing is higher than the central region and the resulting negative lift force is responsible for keeping the leaflets at fully–open position during the forward flow stage. The incoming fully developed pipe flow profile is split into a triple–jet structure by the two leaflets (Fig. 43b). The two side jets near the sinus root are referred to in the literature as the lateral jets while the middle jet is referred to as the central orifice jet. It is clearly seen in Fig. 43 and Fig. 49 that, at this low Reynolds number, the flow in the wake of the leaflets exhibits the quadrant symmetry of the geometry. The dominant flow features of this steady, symmetric flow include the formation of regions of flow reversal and the shear layers downstream of the leaflets and the valve housing as well as the two pairs of counterrotating streamwise vortices that are formed in the wake region.

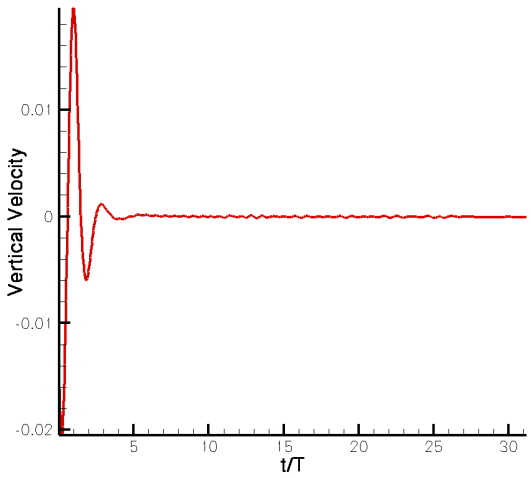
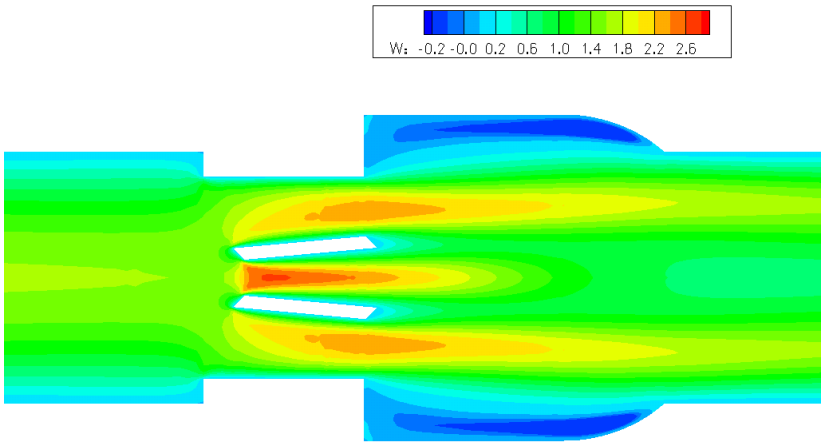
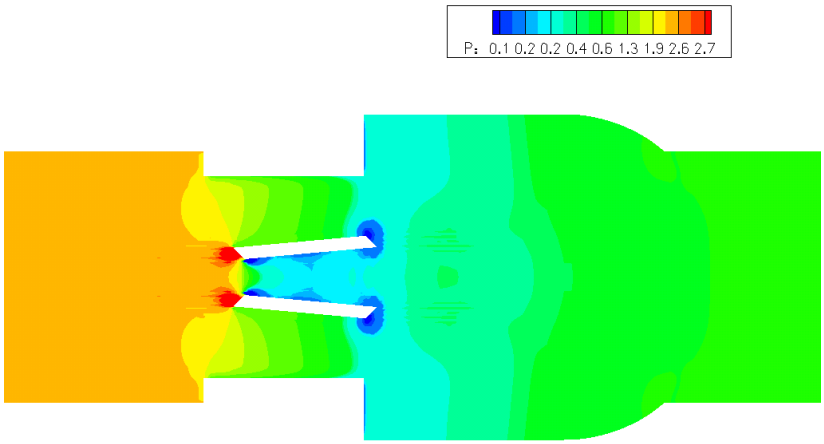


Figure 42: Time history of transverse velocity component at a central point downstream of the valve ($Re = 325$).



(a)



(b)

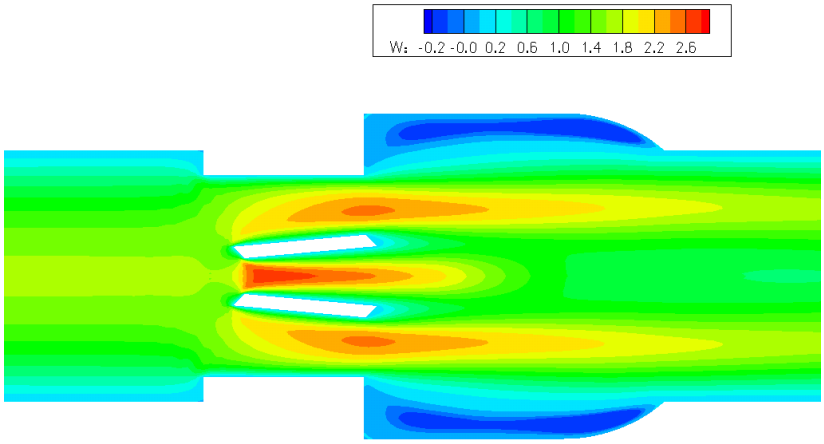
Figure 43: Flow patterns of flow at $Re = 325$: (a) streamwise velocity contours (w/U_0); (b) pressure contours.

Numerical simulations for $Re > 325$ failed to converge to a steady-state solution and weak unsteady modes emerged naturally without any externally imposed forcing other than numerical disturbances in the numerical approach. Fig. 44 shows the calculated instantaneous flow patterns of $Re = 350$. Comparing these figures with those for the Re=325 case reveals very similar overall flow patterns. Yet the time history of the transverse velocity component, which is shown in Fig. 47 (a) at the same point we recorded for Re=325 case, suggests that a weak, albeit persistent over long simulation intervals, unsteadiness appears

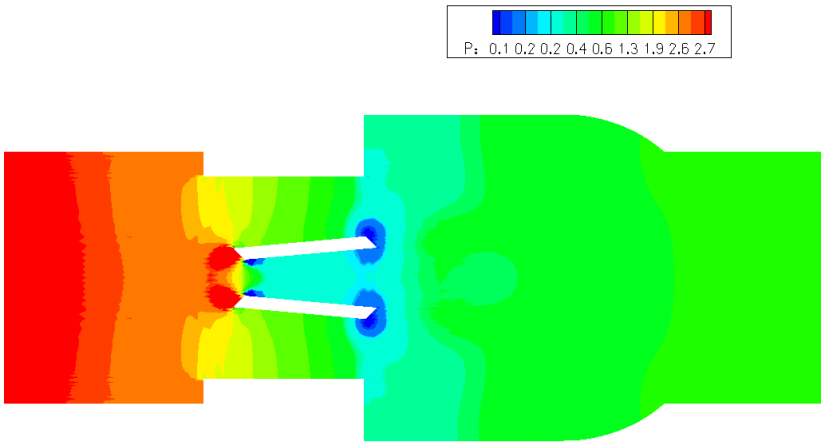
in the wake of the leaflets. The unsteadiness is so weak that the streamwise flow patterns shown in Fig. 44 appear nearly identical to the steady flow patterns for the Re = 325 case. Video animations, however, clearly show the unsteadiness, which manifests itself in the form of a weak streamwise oscillation of the three jets. The jets, however, remain rather stable in the sense that they retain their steady structure and no shedding occurs in the wake of the leaflets.

The intensity of unsteady fluctuations and the richness of the flow were found to increase continuously with Reynolds number. As seen in Fig. 45, the asymmetry of the flow starts to become visible far downstream from the valve in the streamwise velocity contour distributions at the vertical center plane. A better view of the asymmetric flow pattern is provided in Fig. 46, which shows contours of streamwise velocity at horizontal ($y = 0$) symmetry plane—the plane between the leaflets. The instability appears to originate within the shear layer that develops between the jets in the leaflets wake and the annular separation zone in the sinus region downstream of the valve housing. Low amplitude, unsteady modes are excited within the annular separation zone, which interact with the triple jet structure. Consequently unsteadiness develops in the wake, which becomes asymmetric and begins to meanders transversely. It is important to note that in this study unsteady flow sets in at Reynolds numbers considerably lower than those reported in Ge et al. (2003). This trend should be attributed to the different aortic root geometry we employed in this work and underscores the sensitivity of the MHV flowfields to the details of the aortic root geometry.

To quantify the onset of unsteadiness in the flow, we show in Figures 47 and 48 time series and the corresponding power spectrum of the vertical velocity component at the center of a cross-section downstream of the leaflets for Re = 350 and 750, respectively. As mentioned earlier, regardless of whether the flow is steady or unsteady if it preserves the quadrant symmetry of the valve geometry the two transverse velocity components at the center of any cross-section should remain equal to zero at all times. As seen in Fig. 47a, weak unsteady fluctuations about zero emerge for Re=350, which also signify the break of quadrant symmetry. A rather interesting feature that follows from Fig. 47 (a) and the corresponding power spectrum shown in Fig. 47b is the fact that the flow does not

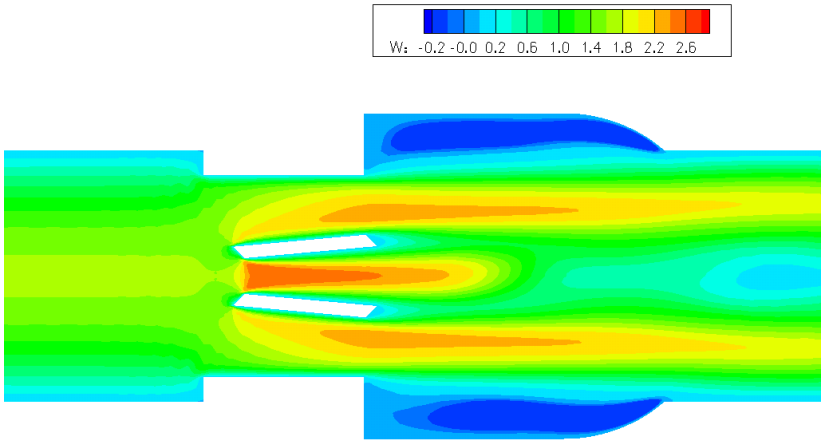


(a)

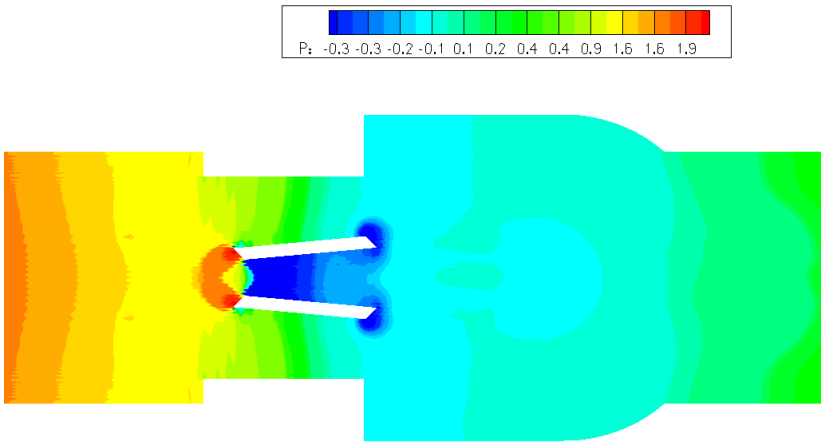


(b)

Figure 44: Snapshot of instantaneous flow patterns at $Re = 350$: (a) streamwise velocity contours (w/U_0); (b) pressure contours.



(a)



(b)

Figure 45: Snapshot of instantaneous flow patterns at $Re = 750$: (a) streamwise velocity contours (w/U_0); (b) pressure contours.

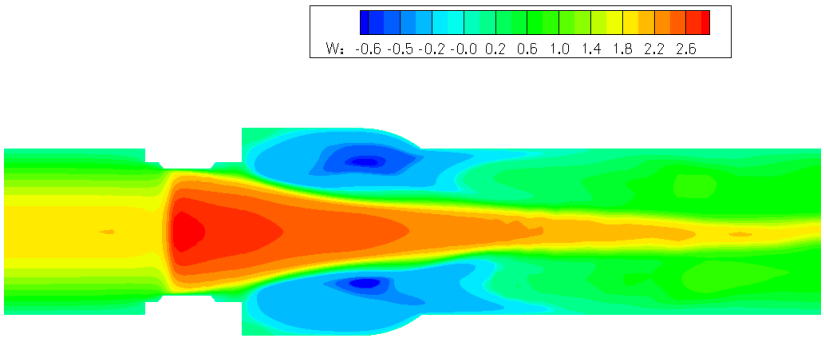
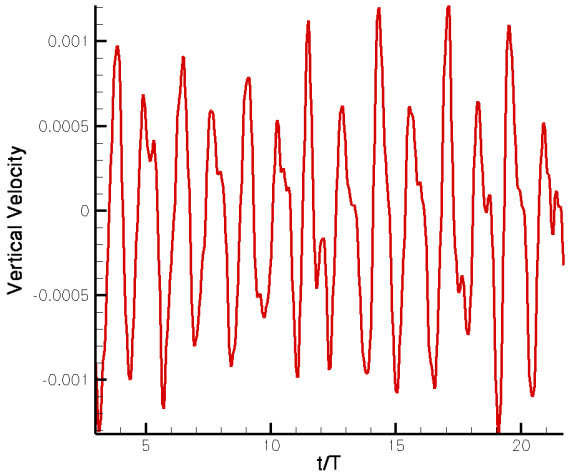


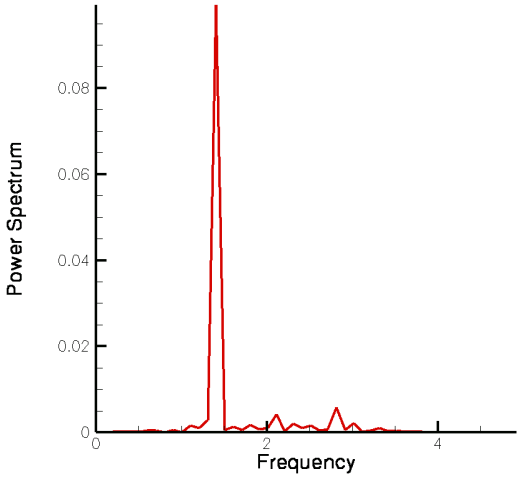
Figure 46: Snapshot of instantaneous flow patterns (streamwise velocity contours (w/U_0)) on $y = 0$ plane at $Re = 750$.

transition to unsteadiness via the emergence of a simple time-periodic mode. Rather the unsteadiness appears to be quasi-periodic in nature with two incommensurate frequencies. The Strouhal number corresponding to these two major dominant frequencies are 0.799 and 1.447, respectively. A similar trend is also observed for Re=750 (see Fig. 48). At this higher Reynolds number, however, both the complexity and amplitude of the fluctuations about the zero mean increase considerably as compared to those for Re=350. As shown in Fig. 48b, the rather complex, chaotic looking time series is dominated by three frequencies. The two basic frequencies, f_1 and f_2, are incommensurate and thus are basic frequencies for this flow. The third frequency f_3 is a harmonic of the two basic frequencies since $f_3 = 2f_1 + f_2$. The Strouhal numbers, St, corresponding to the basic frequencies are 0.466 and 1.521, respectively.

To further elucidate the mechanism that leads to the onset of unsteadiness in the flow, we plot in Figures 49 – 51 a series of instantaneous snapshots of cross-flow vectors, streamwise velocity contours, and limited streamlines at a cross-section downstream of the leaflets for Re = 350, 750, and 1250 . Fig. 49 shows the flow patterns for $Re = 350$ at two different time instants. For this flow, two pairs of streamwise, counter-rotating vortices emerge downstream of the leaflets near the right and left sides of the aorta. The formation of these

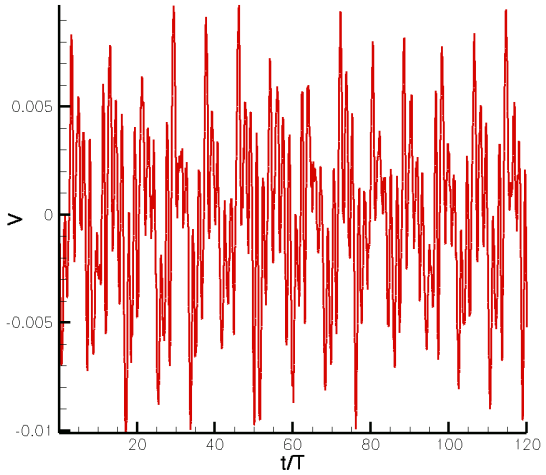


(a) Time history of transverse velocity component at a center point downstream of the valve

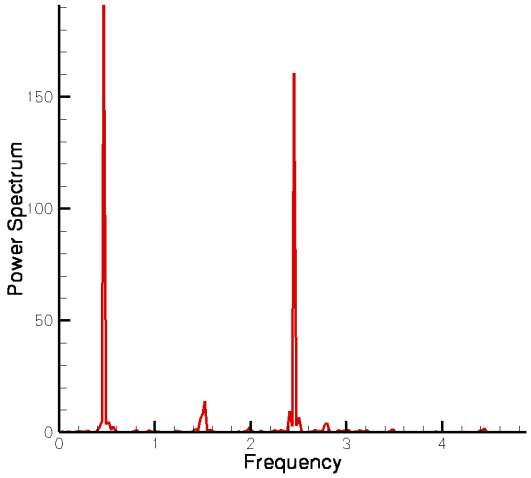


(b) Power spectrum distribution

Figure 47: Time series of flow at $Re = 350$.



(a) Time history of transverse velocity component at a center point downstream of the valve



(b) Power spectrum distribution

Figure 48: Time series of flow at $Re = 750$.

vortices is the result of lateral pressure gradients that develop within the leaflets and the sinus regions, which tend to enhance the production of streamwise direction via the vortex skewing mechanism. These vortices rotate such that their common flow transports low momentum fluid from the aortic wall toward the center of the cross section leading to the characteristic bulging of the isovels in this region. The vortices also act to strain laterally the central orifice jet causing it to switch its major axis from the horizontal to the vertical direction. This important phenomenon will be discussed in more detail subsequently in this section. Comparing the cross-flow vectors at different time instants, it is clearly seen that for $Re = 350$ the two vortical pairs begin to oscillate in time around their fixed mean positions but without altering the basic structure of the flow. For Re=750, however, the strength of the two vortex pairs (vortices A in Fig. 50) increases and they begin to undergo much stronger fluctuations. The unsteady interaction of these vortices with the aortic wall extracts from the wall boundary layer vorticity of opposite sign leading to the formation of new vortices, which appear and disappear in a rather intermittent and seemingly chaotic manner (see Fig. 50b). Flow visualizations show that all these vortices interact with each other, merging and splitting time after time and these interactions are believed to be the reason of the richness of the time series in the wake region. These vortices, which are very strong for Re=750, are anticipated to play very important role in cross–sectional momentum transport. In fact, for the highest Reynolds number simulated in this work (Re=1250), the cross-flow pattern becomes very disorganized with multiple pairs of vortices appearing and disappearing seemingly randomly in time. These vortices cause rather severe deformation of the major jet as suggested by the snapshot shown in Fig. 51.

As discussed above, the vortices that emerge downstream of the leaflets interact with the major orifice jet and cause it to switch its major axis. This complex phenomenon is well documented in the literature for rectangular jets in free-shear flows but has never been discussed before in the context of MHV flows. Fig. 52 shows instantaneous snapshots of streamwise isovels at several cross-sections downstream of the leaflets for Re=750. The snapshot clearly shows the spatial evolution of the main orifice jet from a laterally stretched, rectangular-like jet to a vertically oriented jet. On the area near the valve end (plane F1 on

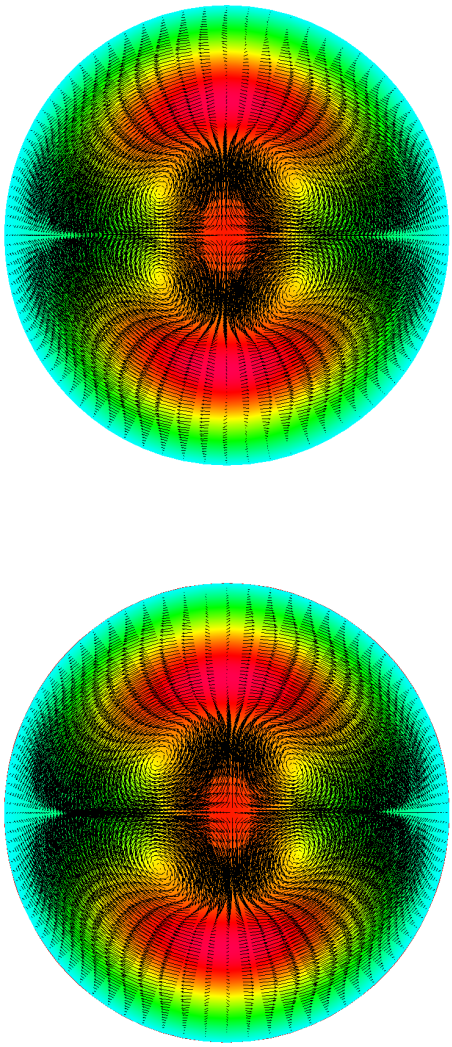


Figure 49: Cross sectional secondary flow patterns at $Re = 350$ (colors show the streamwise velocity contours).

Fig. 52), the rectangular–shaped central orifice jet laterally stretched along the horizontal direction. As the flow developing toward the downstream direction, the streamwise multiple vortices transport low momentum fluid from the near wall region to the center flow area and the major axis of the central jet is clearly compressed (F2 on Fig. 52). In a further downstream plane (F3 on Fig. 52), we can clearly notice that the major axis of the central jet switched to the vertical direction. The central orifice jet ultimately merges with the

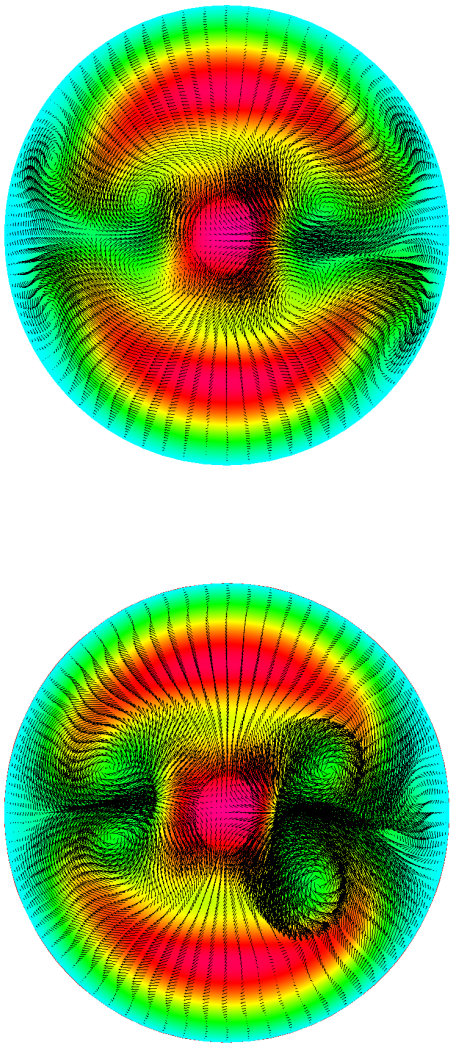


Figure 50: Snapshot of instantaneous cross sectional secondary flow patterns at $Re = 750$ (colors show the streamwise velocity contours).

two lateral jets (F4. on Fig. 52). To the best of our knowledge this is the first time that this complex phenomenon has been documented either computationally or experimentally in MHV flows.

To demonstrate the predictive capabilities of the model we compare in Fig. 53 time-averaged, streamwise velocity profiles for the Re=750 case (for measurement locations see

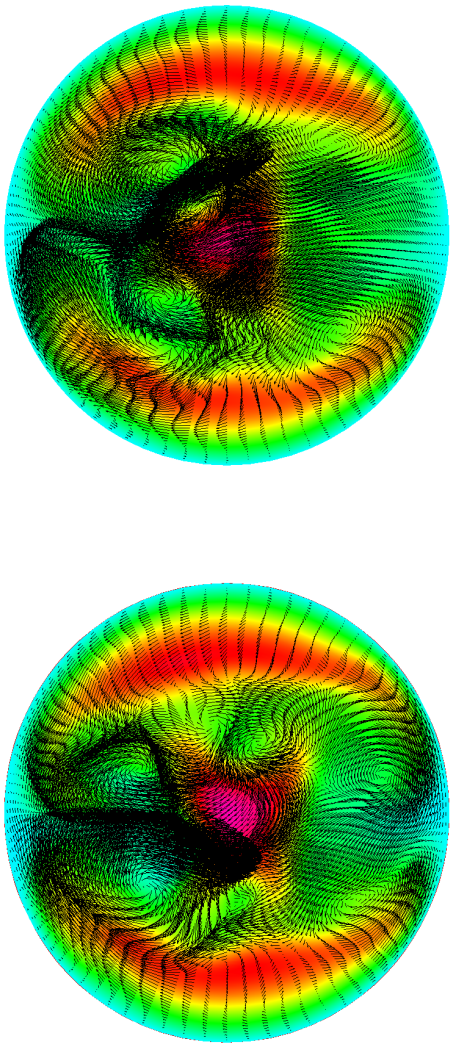


Figure 51: Snapshot of instantaneous cross sectional secondary flow patterns at $Re = 1250$ (colors show the streamwise velocity contours).

Fig. 3). Particle Image Velocimetry (PIV) measurements were obtained at the Cardiovascular Fluid Mechanics Laboratory of the Wallace Coulter School of Biomedical Engineering in a parallel experimental effort aimed at obtaining detailed measurements for validating our numerical model. A description of the experimental set-up and procedures can be found in Ge and Sotiropoulos (2004). Here is suffices to say that the model aorta and valve models used in the experiments are identical to those used in our simulations. We should

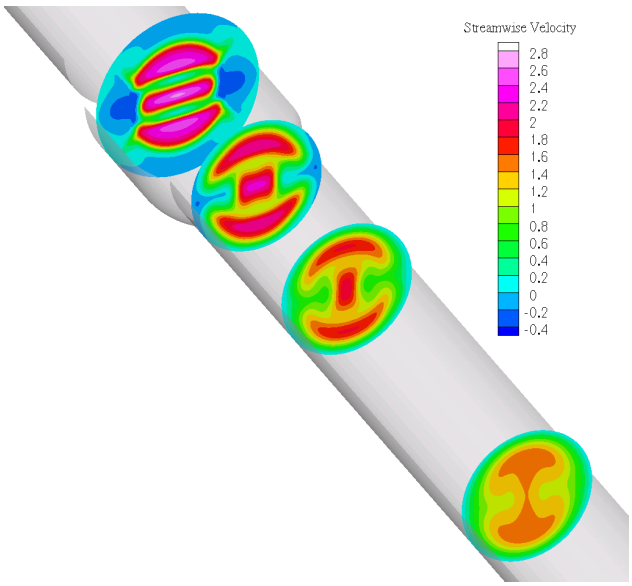
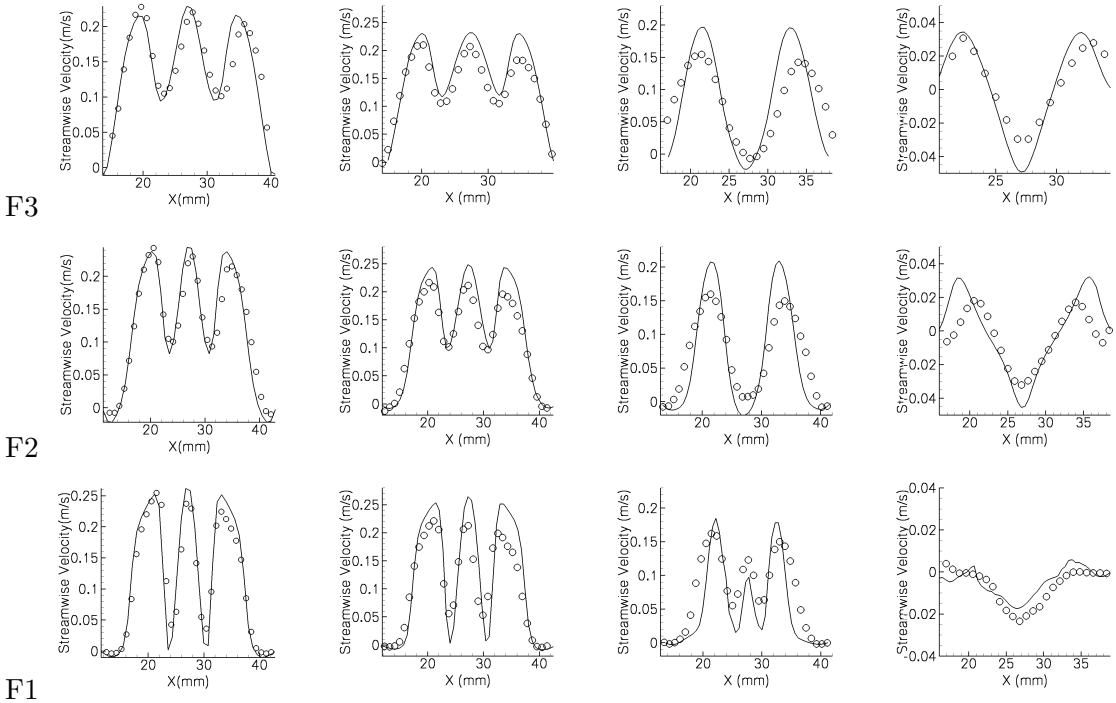


Figure 52: Axis switching of the central orifice jet of flow at $Re = 750$ (colors show streamwise velocity contours).

also mention that during the experiments it was found that the laboratory flow was very sensitive to even very small asymmetries and disturbances and as a result the time-averaged experimental data are not perfectly symmetric. Small asymmetries are evident in the measured streamwise velocity profiles shown in Fig. 53 but as we will subsequently show such asymmetries are even more pronounced for the Re = 6000 case. The asymmetries in the measurements notwithstanding, however, Fig. 53 shows that the numerical simulations capture with remarkable accuracy essentially all features observed in the experiment. Among the most important experimental features that are captured by the computations are the transverse distribution of momentum within the cross-section and the rate at which the three jets diffuse and merge in the streamwise direction.

4.2.2 Turbulent flow regime

Numerical simulations are carried out for Reynolds number of 6000, based on the incoming bulk velocity and the inlet diameter. This specific case is selected to model a peak systolic flow rate of $25L/min$. Flow at this Reynolds number is in the fully turbulent regime and coherent vortex shedding is anticipated in the wake of the leaflets. A grid of similar topology to that shown in Fig. 41 is used to carry out both DES and URANS simulations. The



(a) Center (b) 0.3 R (c) 0.6R (d) 0.8R

Figure 53: Velocity profile comparisons with experimental measurements (seen Fig. 75 for comparison locations).

computational mesh consists of 1.6×10^6 and a dimensionless time step (nondimensionalized by the diameter of the incoming pipe and the bulk incoming velocity) of $dt = 0.01T$ is used for all simulations.

Both the URANS and DES computations yield naturally excited unsteady flows with coherent vortex shedding without introducing any explicit forcing. In order to obtain statistically converged solutions, the DES calculation is advanced in time for 100 time units (which is corresponding to $10,000$ time steps). The URANS calculation is advanced in time for over 50 time units and a statistically converged state is reached. For the entire time interval, the large–scale unsteadiness is sustained in both simulations. In the following section, we are going to present the numerical results as follows: first the instantaneous flow fields are presented and the URANS and DES solutions are juxtaposed with each other. Next, the results from both methods are compared with experimental measurements to validate the simulations. Finally, we present turbulence statistics from the DES and URANS solutions, including power spectra and Reynolds-stress distributions.

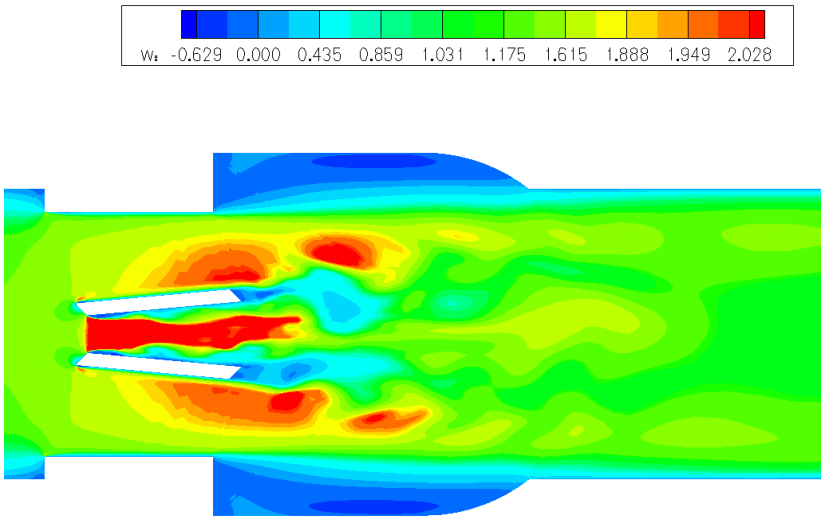
4.2.2.1 Instantaneous flow properties

The instantaneous URANS and DES simulations exhibit highly three-dimensional, unsteady flow patterns with rich dynamics. The complexity of the calculated flow patterns is illustrated in a series of figures, which depict contours, streamlines, and velocity vectors at various two-dimensional planes, including the horizontal and vertical planes of symmetry and a streamwise cross-section downstream of the leaflets.

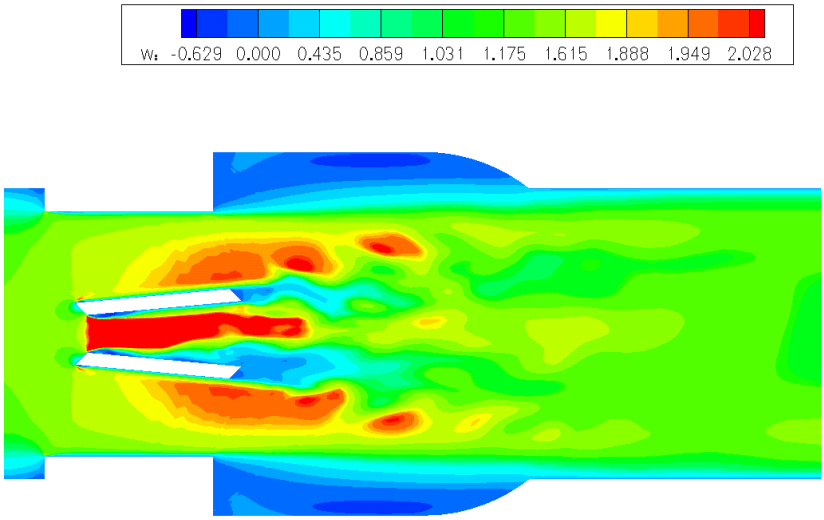
Figs. 54 and 55 each show two snapshots in time of contours of axial velocity calculated with URANS and DES, respectively. The instantaneous URANS solutions are asymmetric and dominated by large–scale unsteadiness. The steady fully developed pipe flow entering the aorta is split into three jets by the two leaflets located in the middle of the main flow lumen. These three jets dominate the flow within the valve region. This triple-jet structure of the flow is broadly similar to the simulated laminar flow patterns discussed in the previous section. The central orifice jet originates from the leading edge of the leeward side of the leaflets facing the downstream direction. The jet separates from the leaflets, forming strong

shear layers between the jet and the boundary layers near the leeward side surface of the two leaflets. The shear layer becomes unstable and undergoes complex temporal oscillations in the wake. The two lateral orifice jets, which were also discussed in the laminar simulations, are also present at this high Reynolds number case. As seen in the figure, the flow in the lateral sides of the leaflets is directed toward the side walls by the leaflets, due to the 85^o opening of the leaflets. The flow in the wake region of the valve is dominated by the complex interaction between the three jets and the annular recirculating flow in the aortic sinus expansion. Two elongated pockets of low streamwise momentum are formed in the trailing edges of the two leaflets between the three jets. The resulting shear layers become unsteady and shed regularly coherent structures in the wake. The instantaneous contours clearly illustrate the complex interaction and coupling of the three jets with the low momentum regions in the leaflets trailing edges. As a result of this interaction and the ensuing Kelvin–Helmholtz instability, pockets of low and high momentum are torn away from the leaflet region and transported by the flow in the wake. The intense unsteadiness in the wake enhances transverse mixing of momentum and as a result the three jets appear to be diffused and mixed approximately one diameter downstream of the leaflets. Note that this intense lateral wake mixing is in stark contrast with the laminar flow patterns in which the triple jet structure was visible for considerable distance downstream of the leaflets.

Another shear zone in the flow originates at the interface between the annular recirculation region in the sinus root and the fast moving lateral orifice jets. Even though the URANS solutions in the main channel are highly unsteady, the flow in that region appears quite smooth and steady. The shear layer appears to remain stable and straight and the flow within the sinus exhibits little if any unsteadiness. This lack of unsteadiness within the sinus region is perhaps the most surprising discrepancy, at least insofar as the streamwise velocity contours at this plane are concerned, between the URANS and DES predictions. As seen in Fig. 55, the DES flowfields exhibit essentially the same overall patterns with the three jets becoming unstable and mixing rapidly downstream of the leaflets. Overall DES resolves smaller scale structures and yields a more complicated wake flow. This finding is entirely consistent with the fundamentally different modeling philosophies in the two

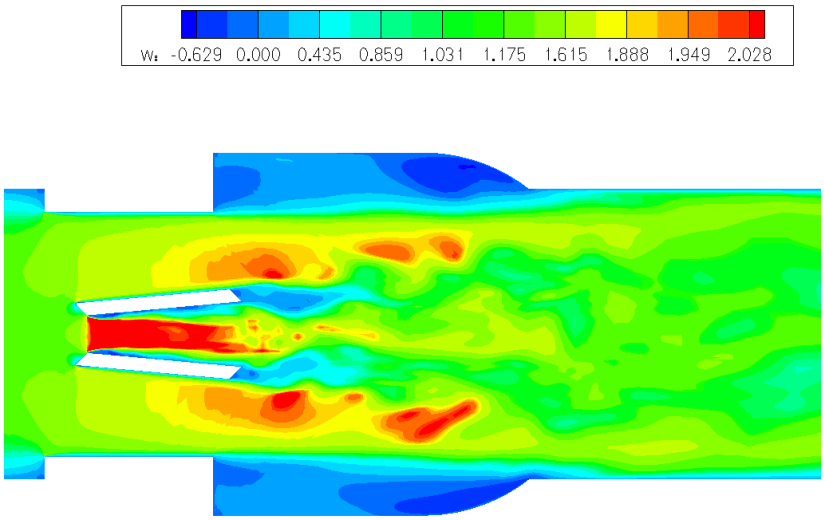


(a) instantaneous time 1

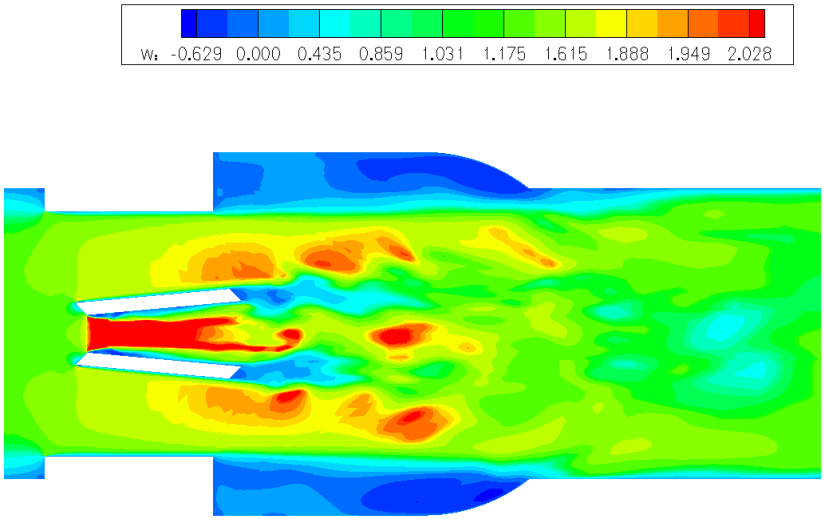


(b) instantaneous time 2

Figure 54: Instantaneous streamwise velocity contours on x=0 plane ($Re = 6000$, URANS solutions)



(a) instantaneous time 1

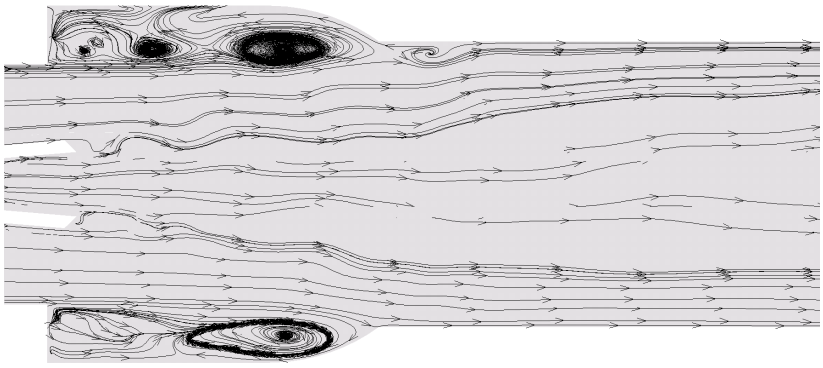


(b) instantaneous time 2

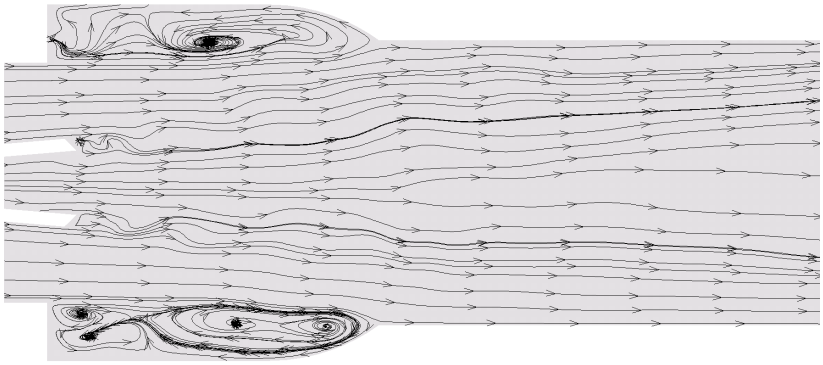
Figure 55: Instantaneous streamwise velocity contours on x=0 plane (Re = 6000, DES solutions)

models. In the URANS approach, the model is designed to capture only the largest scale coherent motion and the eddy viscosity is rather dissipative. All small scale flow features are inevitably smoothed out and this is quite evident by comparing Figs. 54 and 55. In the DES approach, on the other hand, in regions far from solid walls the model uses a characteristic length scale proportional to the local grid spacing, thus, transitioning to the far less dissipative LES mode. Consequently, and for a given spatial and temporal resolution, it is to be expected that DES will resolve directly a larger portion of the turbulence structure than URANS. Within the sinus region the DES flowfields suggest that the recirculation zone exhibits low frequency temporal oscillations and the shear layer is wavy and unstable. At first glance and as mentioned before, this finding is a bit surprising since within this region the flow is essentially laminar and there is no obvious region why URANS would not be able to predict similar unsteadiness. Note, however, that the apparent unsteadiness of the flow in the sinus is driven by the instability of and interaction with the lateral orifice jet shear layers. Obviously the more dissipative URANS can not excite this more subtle instability and, thus, the sinus region flow remains stable and steady.

To elucidate the drastically different flow patterns obtained by the two approaches in the sinus region, we show in Fig. 56 and Fig. 57 2D instantaneous streamlines obtained with DES and URANS. As concluded by examining the velocity contours, URANS yields a steady sinus flow dominated by a single recirculating eddy. DES on the other hand yields very complex patterns with multiple eddies appearing and disappearing in a seemingly random manner within the sinus region. Animations show that the center of these vortices oscillate in time. At certain time instants, small eddies are shed from the downstream end of the recirculation region and are advected downstream along the aortic wall as shown in Fig. 57a. Because of the limited resolution of the current experimental data, we are not able to make detail comparisons of these flow strucutres with experimental data at current time. However, animations of the experimental data do show the existence of unsteady flow structures in the separation zone. In a recent obtained experimental data set with much higher resolution (geometries of the valve are different from the current numerical modeled one), such features are clearly shown in the separation zone (Fig. 58).

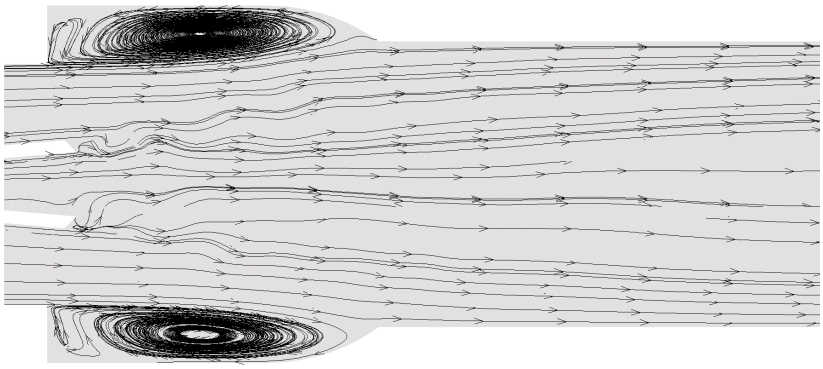


(a) instantaneous time 1

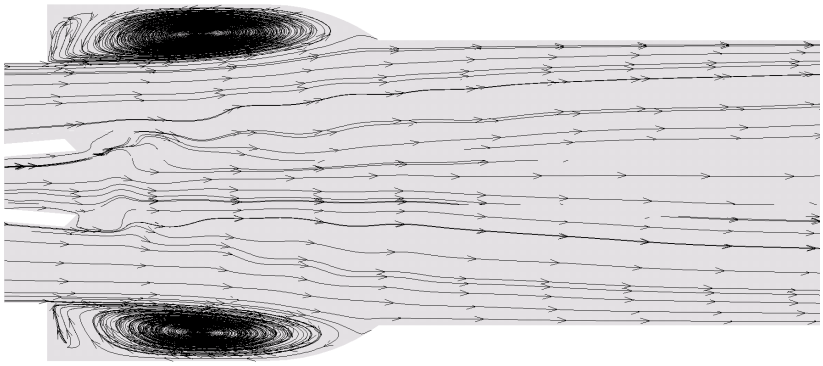


(b) instantaneous time 2

Figure 56: 2D limited streamtraces on $x = 0$ plane ($Re = 6000$, DES solutions)



(a) instantaneous time 1



(b) instantaneous time 2

Figure 57: 2D limited streamtraces on $x = 0$ plane ($Re = 6000$, URANS solutions)

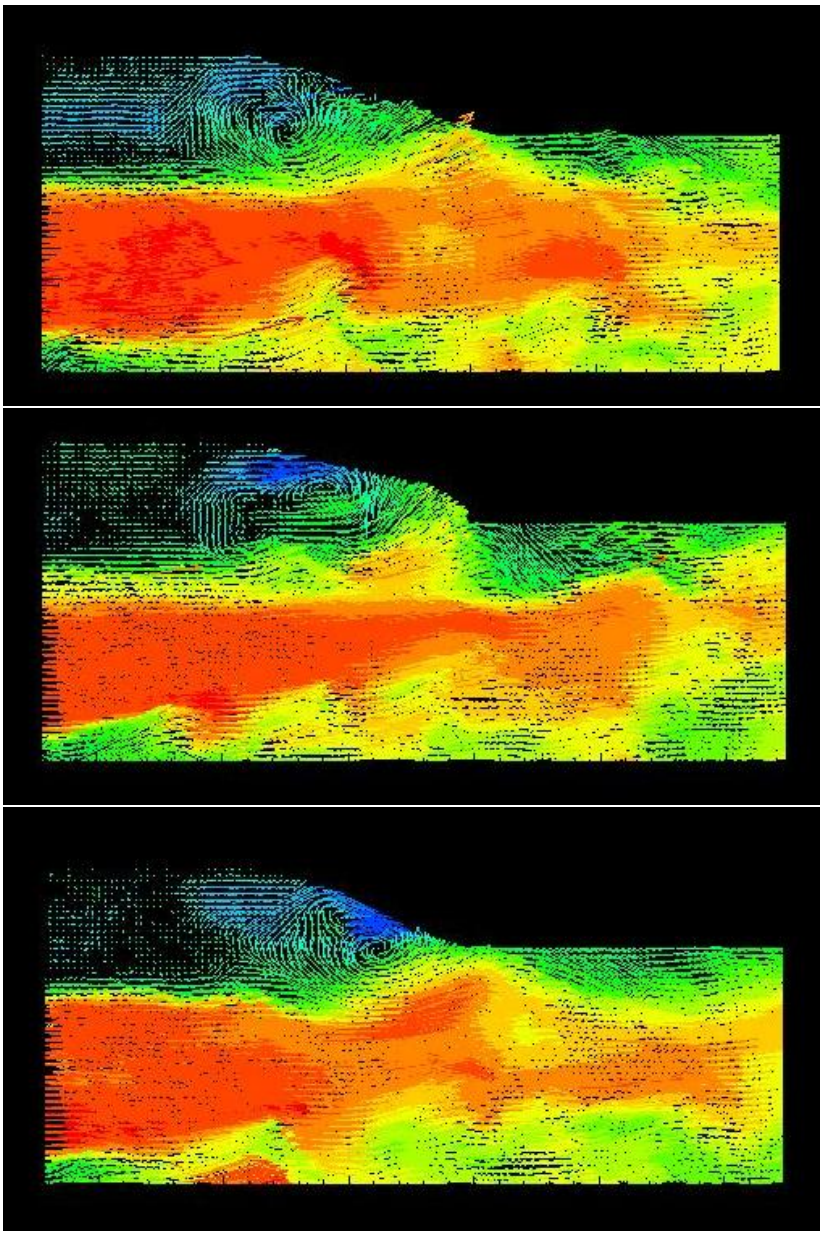
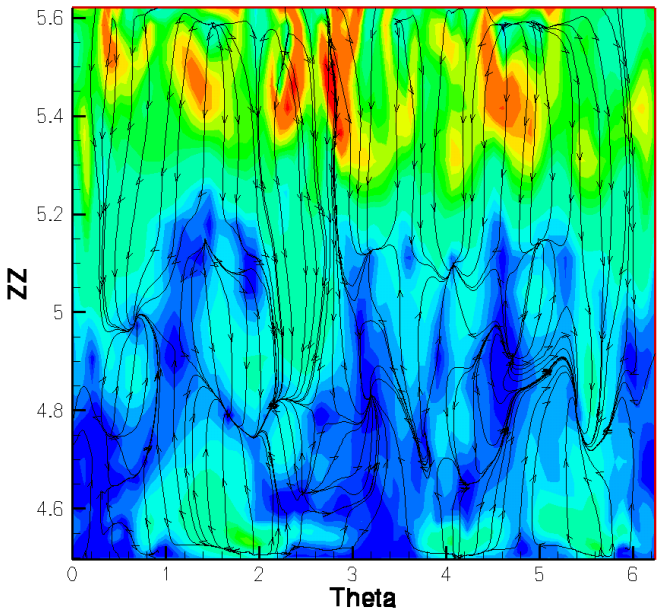


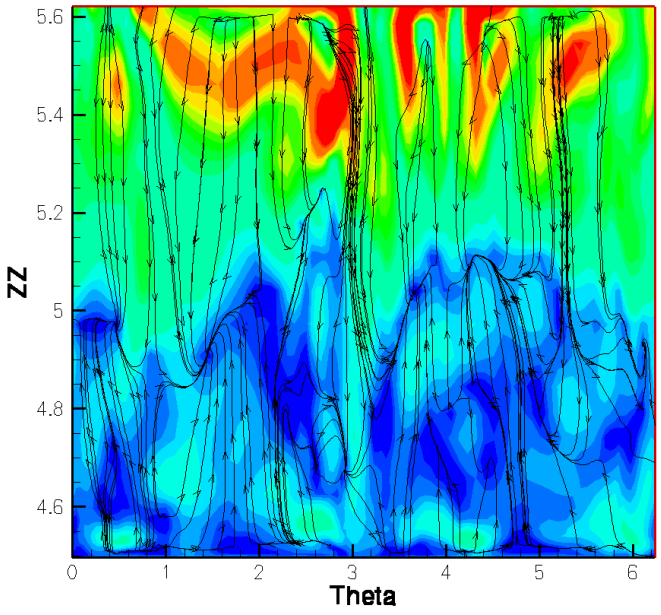
Figure 58: Coherent vortical structures observed in recent experiments

The DES 2D streamline plots suggest that the sinus flow is very complex and highly three-dimensional and unsteady. To further clarify the structure of the DES flowfields in this region we show in Fig. 59 instantaneous limiting streamlines along the aortic sinus wall. To construct these figures we have unfolded the sinus wall and plotted it in terms of the azimuthal arc length and the streamwise coordinate. The limiting streamlines have been constructed using the near wall velocity field projected in the circumferential and axial directions along the sinus wall. Instantaneous contours of axial velocity at the grid-surface just off the sinus wall are also included in the figure as they provide an estimate of the skin friction magnitude. As seen at the downstream end of the plot, which corresponds to the downstream edge of the sinus root, the fluid flows backward toward the upstream direction (incoming flow direction). This flow direction is consistent with the existence of recirculating flow within this region. Note, however, that for flow with axial symmetry the recirculating flow in this region would be a toroidal structure and its projection in the limiting streamlines plot would be a straight line of convergence (from left to right in this plot), which would delineate the forward and backward flow. Yet, the limiting streamline patterns shown in the figure suggest a far more complex flow pattern with distinct azimuthal modes developing in the limiting streamlines. These modes cause the horizontal line of convergence to become wavy and develop a complex topology with multiple singular nodes such as saddles and foci. Animations further show that the near wall flow within the sinus is very unsteady and is dominated by the continuous meandering in the azimuthal direction of the pockets of high positive shear stress at the downstream end of the aortic sinus. It is important to point out that this is the first time that such complex flow patterns are uncovered for the near wall flow in the sinus. The oscillatory nature of the limiting streamlines and the pockets of high shear at the downstream end could have important implications for the bio-chemistry of the endothelial cells lining the aortic vasculate and thus deserves further investigation.

To quantify the significant differences between the instantaneous URANS and DES flowfields in the sinus region, we plot in Figs. 60 and 61 the velocity time–history at two points, A & B, and the corresponding power spectrum distribution. Point A is located on the valve center line, two diameters downstream of the end of the leaflets, while point



(a) instantaneous time 1



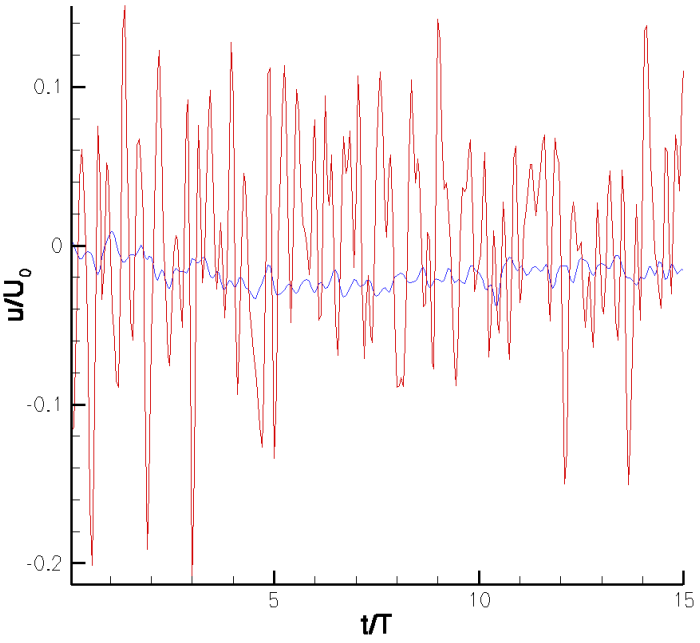
(b) instantaneous time 2

Figure 59: Limited streamtraces near sinus wall of DES solution

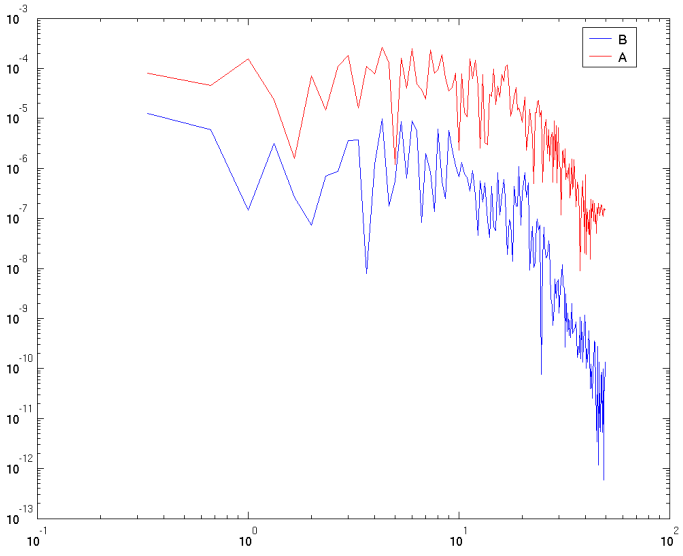
B is located in the recirculating aortic sinus flow. As seen in these figures, the energy of the flow at this Reynolds number is distributed at a rather wide range of frequencies. No specific dominant frequency can be picked from the power spectrum distribution. In the central flow region, (Point A) the DES and URANS power spectra are similar, with the DES yielding overall richer dynamics. On the other hand, the time series at point B are drastically different. In the DES solution the power at point B is orders of magnitude higher than of the URANS solution and at the same level as that at point A. In fact at low frequencies, the DES power at point B is higher than that at point A. This finding confirms that in the DES flowfield, the sinus region is dominated by low-frequency coherent structures whose dynamics is very complex.

The instantaneous flow patterns on the horizontal ($y = 0$) plane of symmetry are shown in Figs. 62 and 63. Both figures show snapshots of streamwise velocity components at the same instant in time as those shown in Figs. 54 and 55. A notable feature of the flow in this plane is the strong lateral compression of the main orifice jet, which leads into the observed triangular shape of the iso-vels downstream of the leaflets. Both URANS and DES capture this feature of flow but the latter yields, as anticipated, a far more complex wake flow with intense unsteadiness and shedding. The strong interaction between the major orifice jet and the recirculating region in the sinus is also evident in the DES results. As seen in Fig. 63 the central orifice jet meanders laterally on that plane and acts to destabilize and disorganize the sinus flow. This complex interaction between the sinus flow and the main orifice jet is obviously not captured by URANS, which yields a much more stable central jet.

As became evident from the laminar flow simulations, the cross sectional secondary flow plays a very important role in momentum transfer with the cross sectional plane. In Fig. 64 we show an instantaneous snapshot of the cross sectional flow calculated with URANS. The cross sectional plane shown here is located one diameter downstream of the leaflets. Overall, the secondary flow consists of multiple streamwise vortices which are located asymmetrically within the cross section. Animations show that the cores of these vortices undergo low amplitude temporal oscillations within an otherwise relatively stable cross-sectional ambient



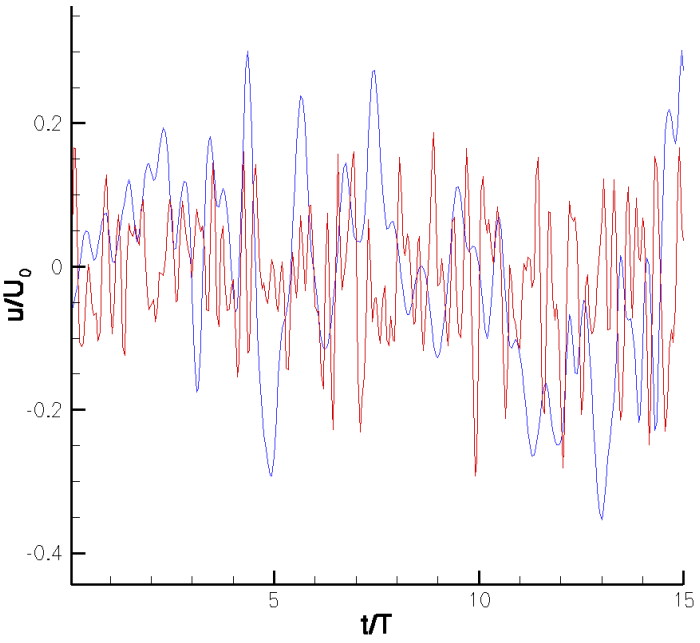
(a)



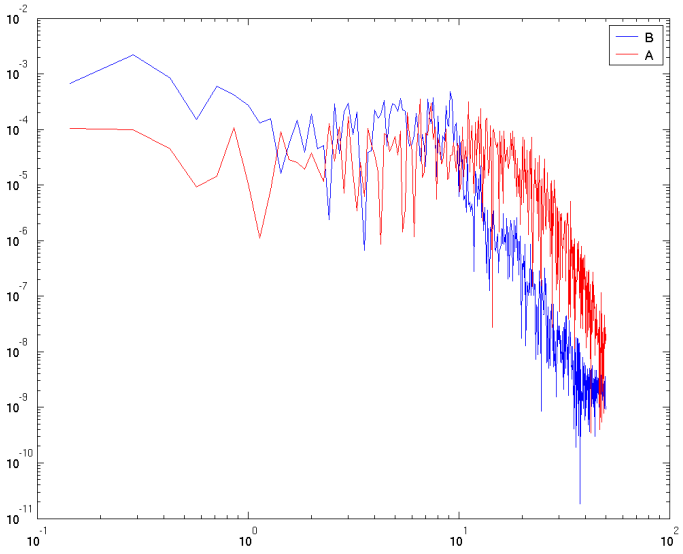
(b)

Figure 60: (a) Time history of u and (b) power spectrum of URANS solutions ($Re = 6000$)

. The sense of rotation of these vortices is consistent with the existence of low streamwise momentum regions near the wall along the horizontal diametrical plane (Fig. 64 a). The



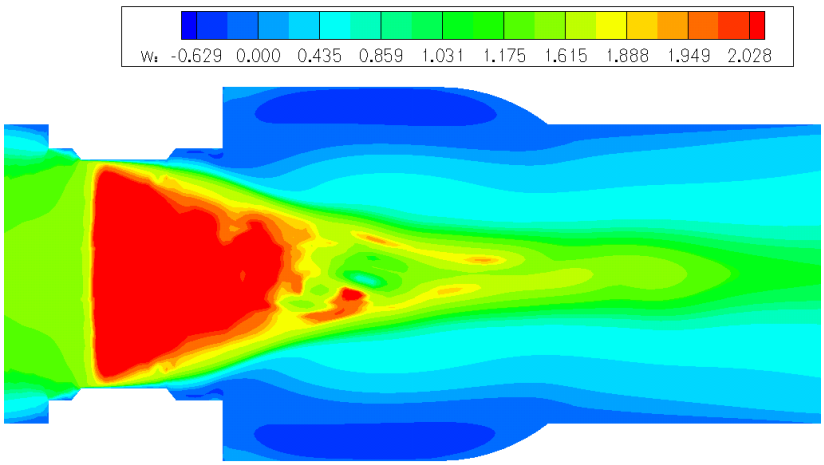
(a)



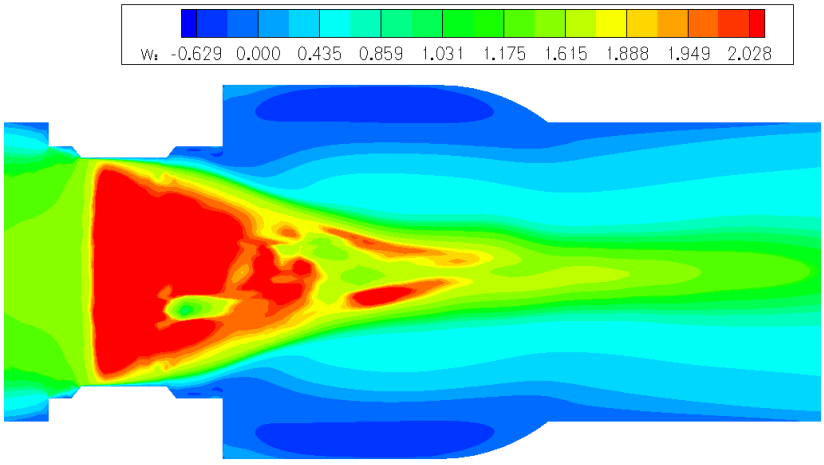
(b)

Figure 61: (a) Time history of u and (b) power spectrum of DES solutions ($Re = 6000$)

triple–jet structure in the flow is still visible but a new feature of the flow at this high Re, which was not there in the laminar flow simulations, is the appearance of two pockets of

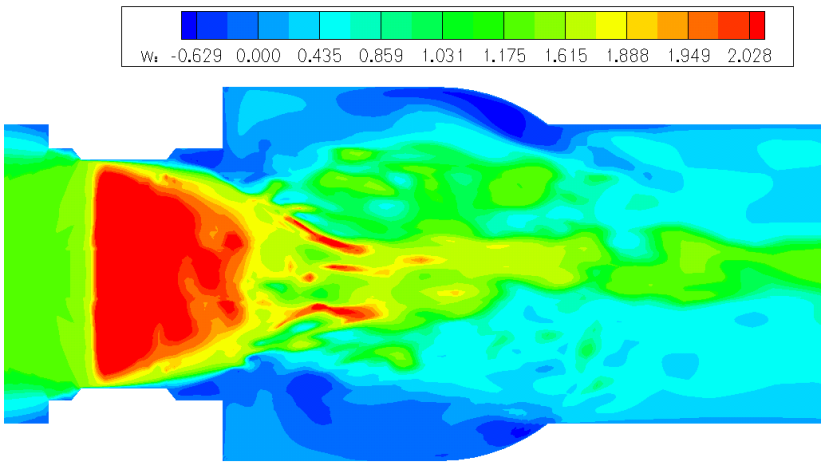


(a) instantaneous time 1

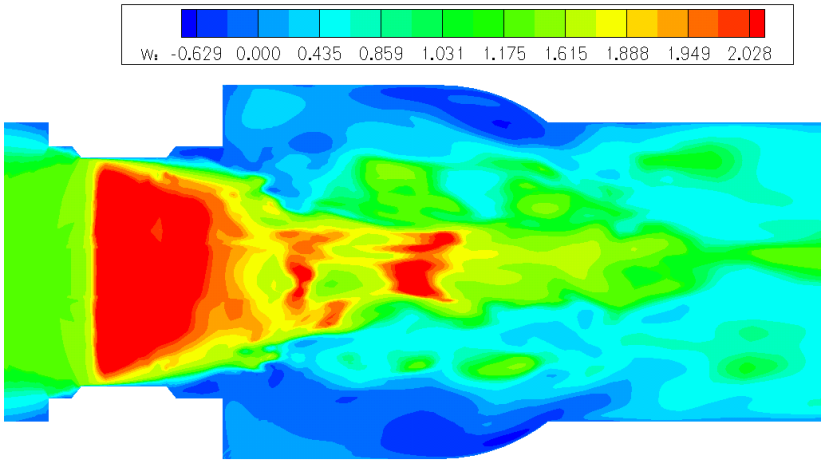


(b) instantaneous time 2

Figure 62: Instantaneous streamwise velocity contours on y=0 plane (URANS)



(a) instantaneous time 1



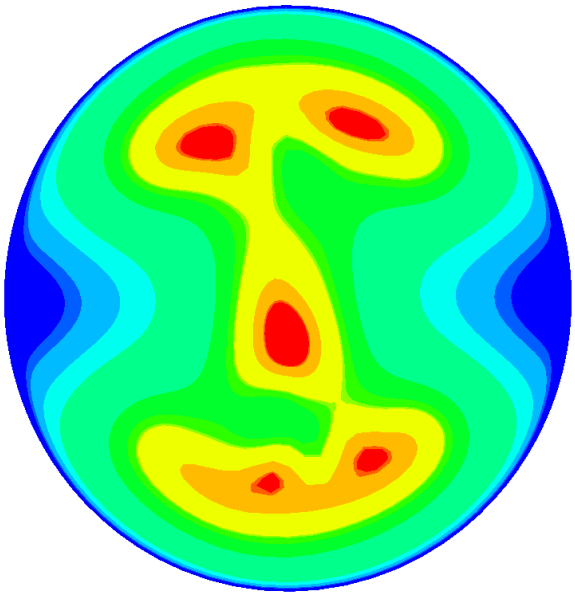
(b) instantaneous time 2

Figure 63: Instantaneous streamwise velocity contours on y=0 plane (DES)

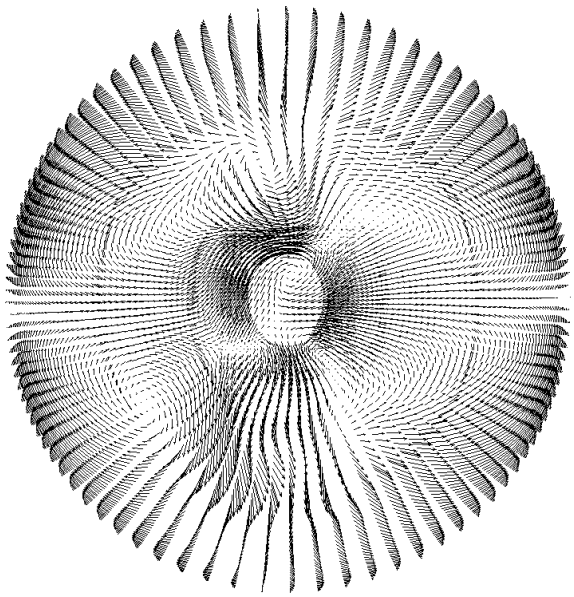
high streamwise momentum within the lateral orifice jets. On the cross sectional vector figure (Fig. 64 b), we can observe several streamwise vortices in this plane. A dramatically different picture for the instantaneous cross-sectional flow emerges from the DES results shown in Fig. 65. In contrast with the organized flow patterns in the URANS solution, the DES solution is dominated by: a) multiple, randomly distributed pockets of high velocity flows; b) multiple pockets of low streamwise momentum distributed randomly along the circumference of the aortic wall; and c) multiple and very intense streamwise vortices. Animations of the instantaneous cross sectional secondary flow pattern show that, unlike the weak unsteadiness in the URANS solution, these vortices undergo intense and very complex temporal oscillations and they appear and disappear rather chaotically. At the cross-section shown in this figure, the triple–jet structure is entirely disorganized and can hardly be distinguished from the instantaneous velocity contours.

Snapshots of instantaneous vorticity contours ω_x at the $x = 0$ plane of symmetry are shown in Fig. 66 for URANS and Fig. 67 for DES. These results re-enforce the previously discussed trends. In the URANS flowfields, unsteady, coherent vortex shedding is only observed in the vicinity of the leaflets while in the sinus region the flow and shear layers are quite stable. On both sides of each leaflet, the elongated shear layers are clearly marked by pockets of high vorticity and extend all the way to the wake. After about 0.3D downstream of the leaflets, the shear layers break up into smaller, disorganized vortices. The growth of the shear layer at the interface between the main channel flow and the sinus flow is also clearly illustrated in these two snapshots. On the side where the valve housing and the sinus root intersect, the vorticity magnitude is maximum but the URANS shear layers in this region remain stable and steady. The DES vorticity contours shown in Fig. 67 emphasize the ability of DES to capture very rich vortical dynamics in the wake region. They also show very clearly the instability of the shear layers near the sinus and their complex interaction with vorticity of opposite sign shed from the leaflets.

To underscore the highly three-dimensional structure of the flow, we show in Figs. 68 and 69 two instantaneous iso–surfaces of helicity density obtained with URANS and DES,

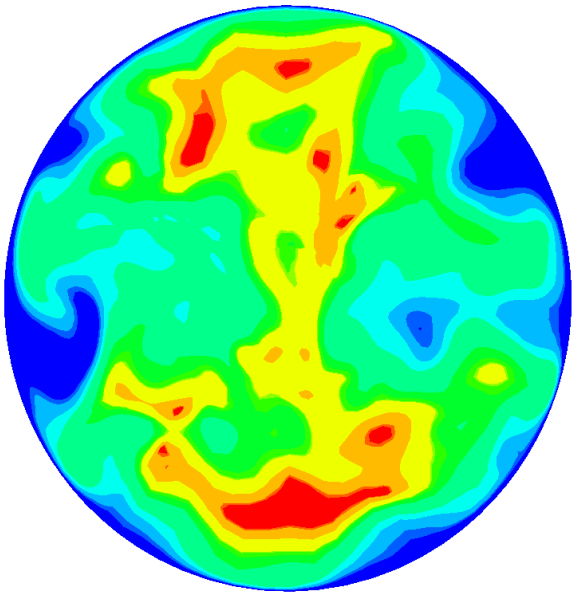


(a)

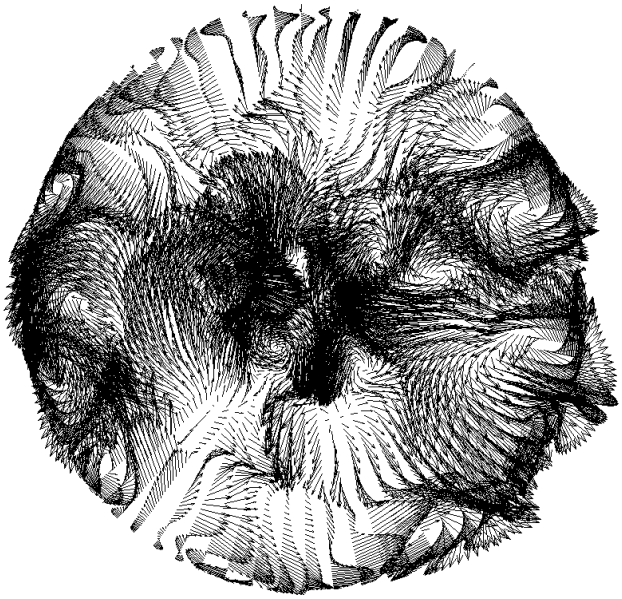


(b)

Figure 64: Instantaneous cross sectional secondary flow patterns of URANS solution ($Re = 6000$)

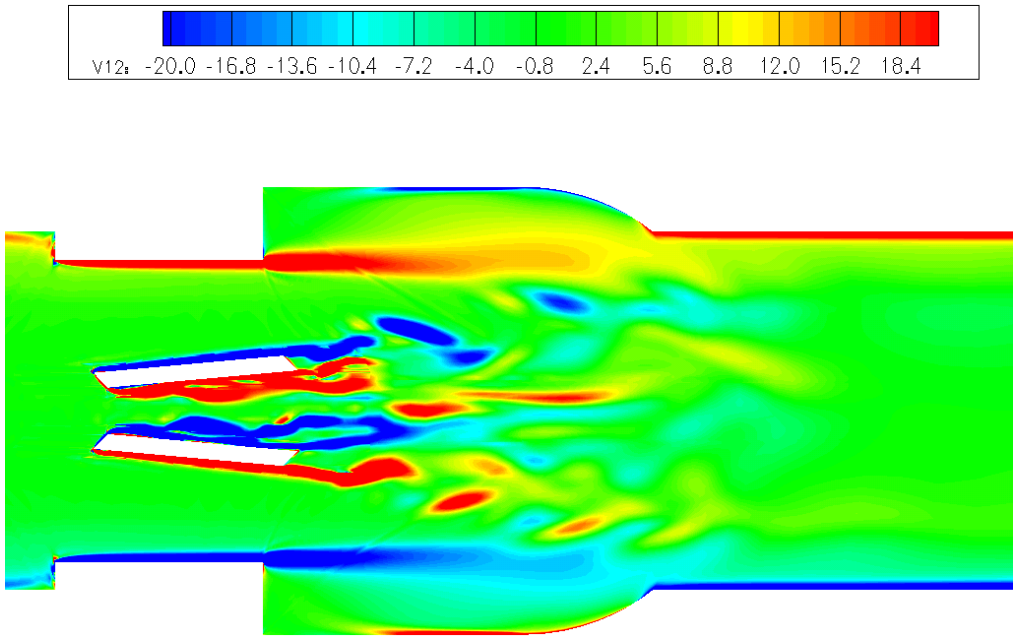


(a)

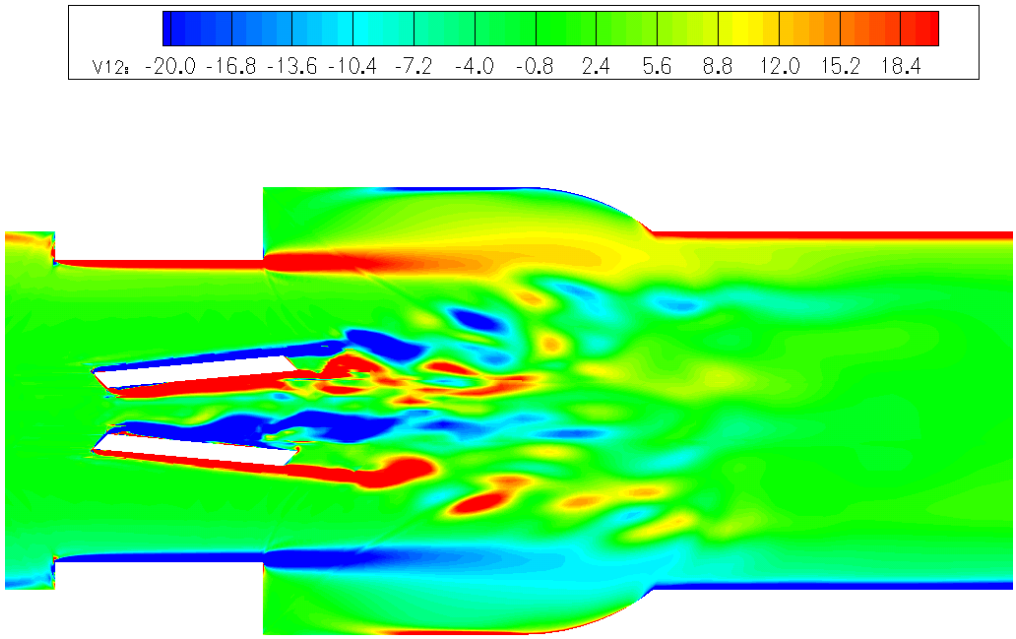


(b)

Figure 65: Instantaneous cross sectional secondary flow patterns of DES solution ($Re = 6000$)

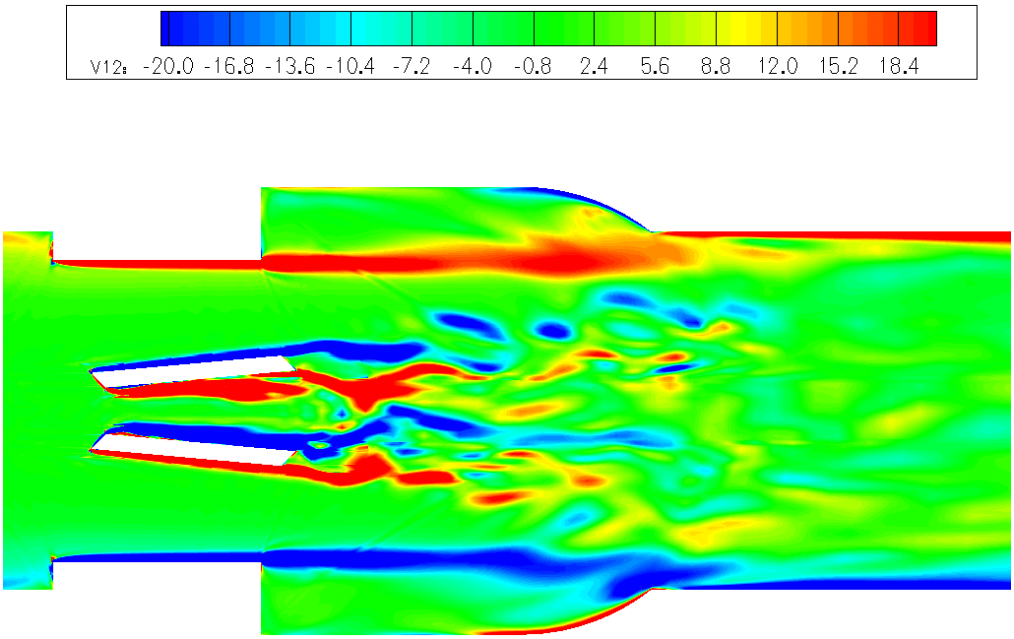


(a) instantaneous time 1

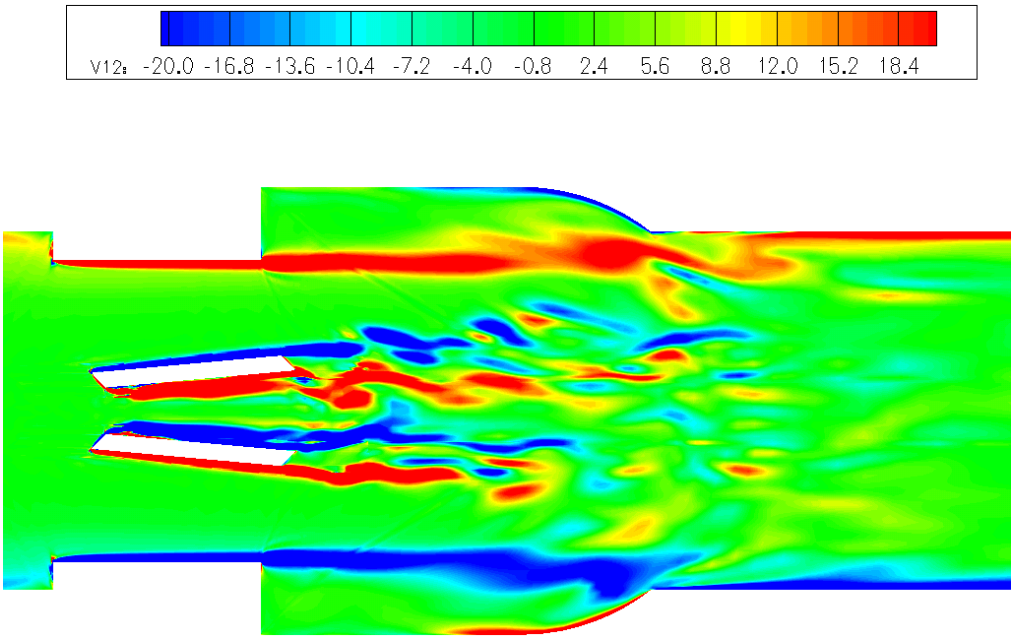


(b) instantaneous time 2

Figure 66: Instantaneous vortex contours (Ω_x) on x=0 plane (URANS solutions)



(a) instantaneous time 1



(b) instantaneous time 2

Figure 67: Instantaneous vortex contours (Ω_x) on x=0 plane (DES solutions)

respectively. Helicity density is defined as the dot product of the velocity and vorticity:

$$H = \bar{U} \cdot \bar{\omega} \tag{63}$$

where H is the helicity, $\bar{U}$ the velocity vector and $\bar{\omega}$ the vorticity vector, and can be used to identify coherent vortical structures where vorticity and velocity are aligned. As shown in Fig. 68, in the URANS flowfield counter-rotating, coherent streamwise vortices emanate from the surfaces of the leaflets. These vortices are stretched by the streamwise pressure gradients and are seen to form characteristic hairpin-like loops in the wake. These features are clearly visible in the URANS solutions but the DES iso-surfaces are so complex and rich that complicate any attempt for rational interpretation. Yet they too reveal the presence of multiple streamwise vortices and show some evidence of hairpin-like loops in the wake of the leaflets.

Overall, the results shown in this section are rather striking and revealing. They clearly illustrate the enormous richness of the vortical dynamics captured by DES at a computational cost essentially identical to that of the URANS approach.

Finally, we should point out that due to the specific materials used for heart valve manufacturing, the intra–valvular flowfields are quite difficult to study experimentally and thus considerably less is know about this region of the flow than the more widely studied wake flow region. In Subramanian et al. (2000), a transparent bileaflet valve was designed to investigate the intra–valvular flow pattern under pulsatile flow condition. The instantaneous velocity and vorticity contours they obtained from their experiment at peak systole condition are shown in Fig. 70. It is very encouraging to note that in spite of significant differences between the experimental conditions and our simulations—the most important of course being that in the experiment pulsatile flow was studied—both the URANS and DES results exhibit broadly similar qualitative features with the experimental data. Specific features that are similar include the general structure of the central and lateral orifice jets and the presence of pockets of low streamwise momentum at leaflets trailing edges. A more rigorous validation of our computations is given in a subsequent section.

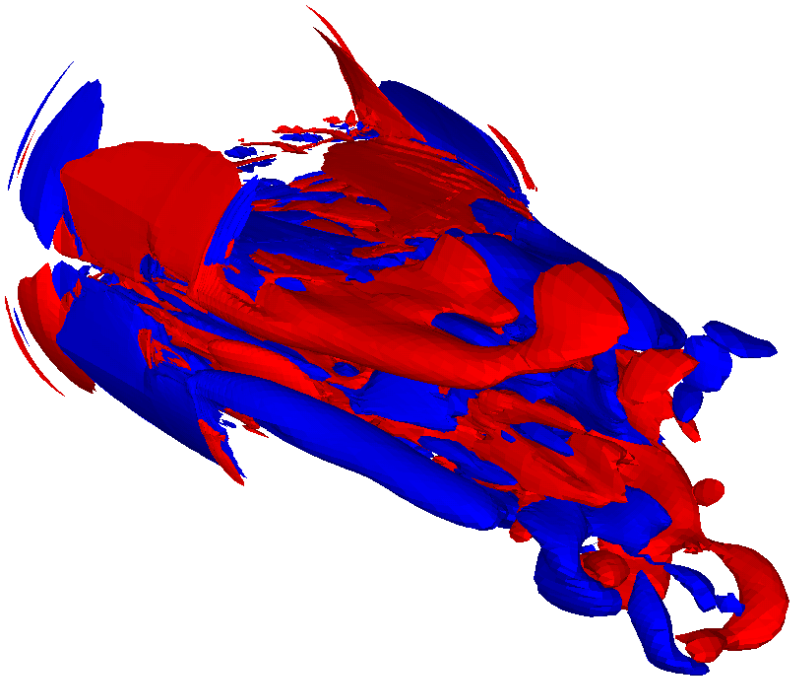


Figure 68: Isosurfaces of helicity ($Re = 6000$, URANS solutions). Contour levels: Red $H = +5$, Blue $H = -5$

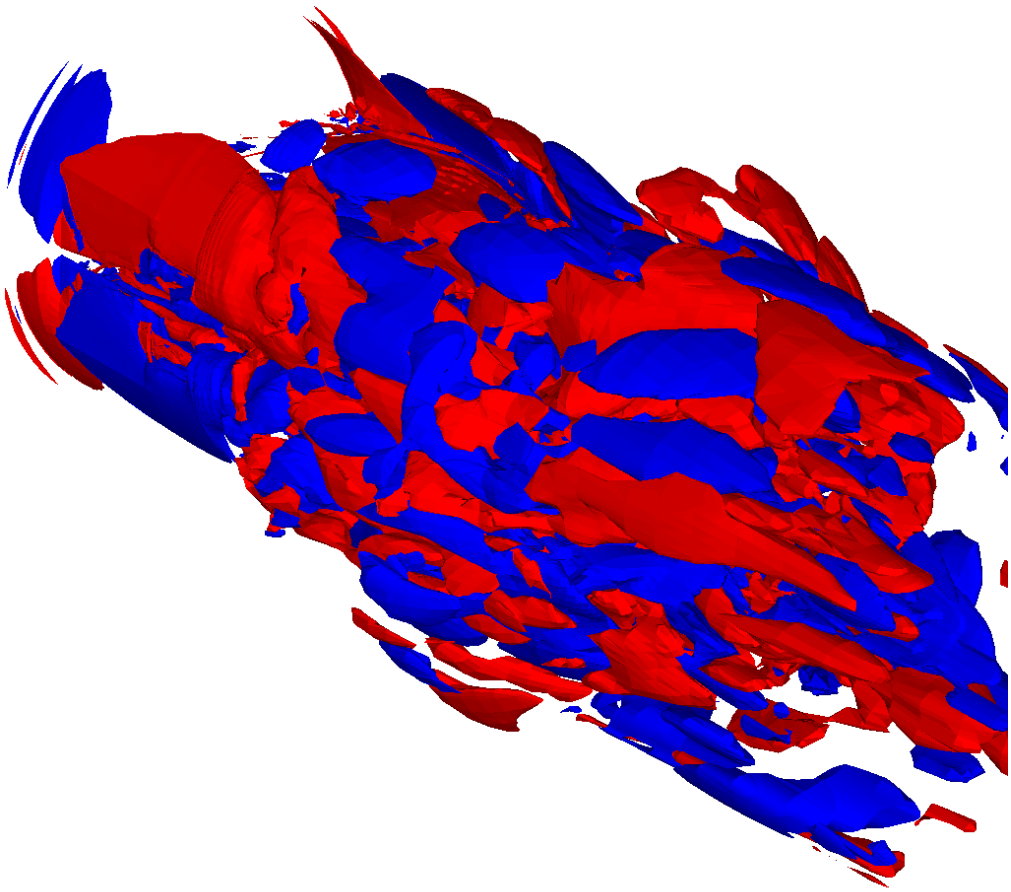


Figure 69: Isosurfaces of helicity ($Re = 6000$, DES solutions). Contour levels: Red $H = +5$, Blue $H = -5$

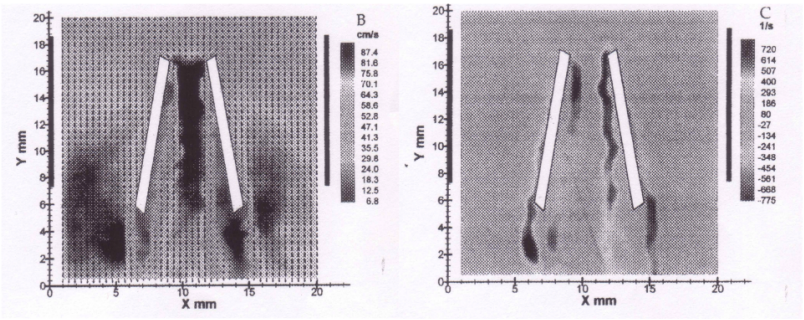
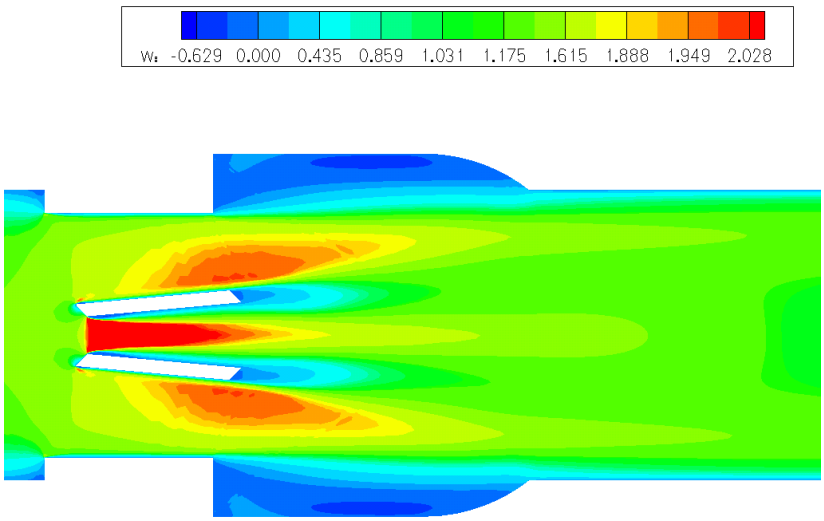


Figure 70: Experimental observation of velocity and vorticity contours at systole peak (Subramanian et al., 2000)

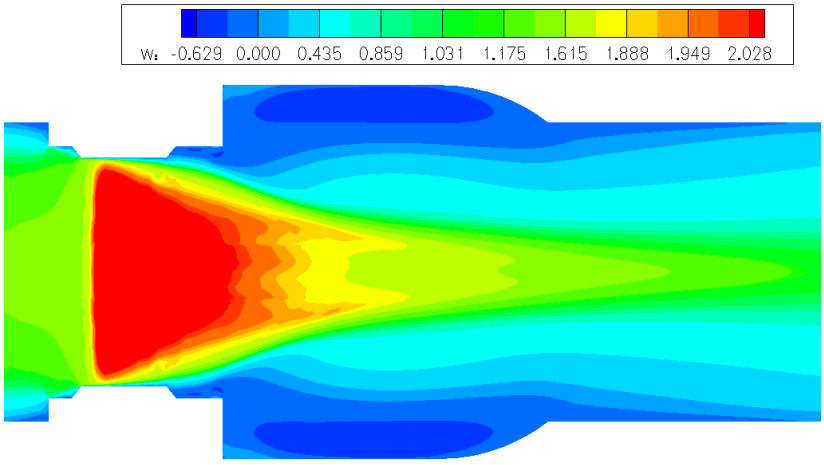
4.2.2.2 Time–averaged flow field

The time averaged streamwise velocity contours are shown in Figs. 71 and 72 for the URANS and DES, respectively. Overall the time–averaged solutions obtained from these two different approaches are quite similar, except for the pocket of recirculating flow in the sinus region. The triple-jet structure, which is characteristic of bileaflet heart valve flows is evident in both simulations at the vertical plane of symmetry. The highest streamwise velocity appears at the central orifice jet with a value of approximately 2.4U (U is the bulk velocity of the incoming flow) at about 0.1D from the leading edge of the leaflets. At the lateral sides of the leaflets, the mean flow accelerates and reaches a local velocity peak of about 2U near the wake tip of the leaflets. Although the local velocity peak appears to be higher in the central flow region, the two lateral orifice jets extends further downstream than the central jet. The DES yields smaller maximum velocity in the lateral orifice jets than that computed with URANS. This trend, however, is consistent with the fact that DES yields a smaller and weaker pocket of recirculating flow within the sinus. This is an important difference between the two approaches and will be discussed in more detail below.

The time-averaged cross-sectional flow is shown in Figs. 73 and 74 for the URANS and DES, respectively. The URANS time–averaged flow field doesn't differ drastically from the instantaneous flow field shown in Fig. 64, suggesting that the instantaneous URANS

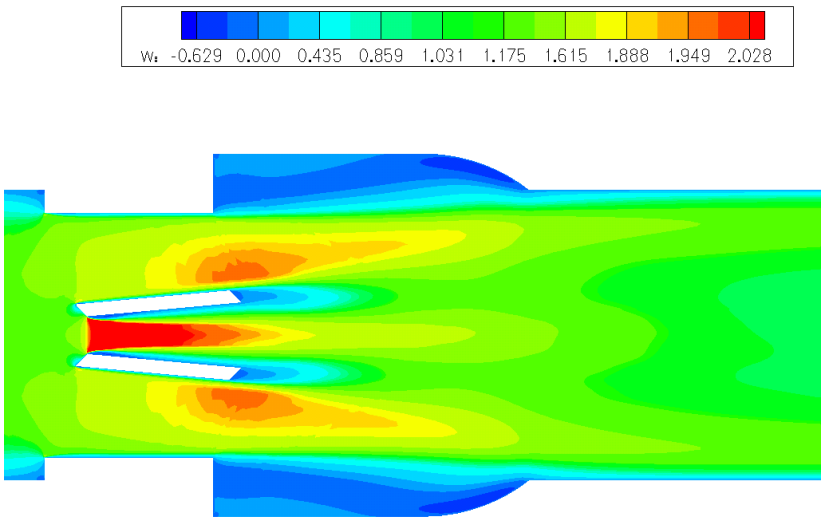


(a) $x = 0$ plane

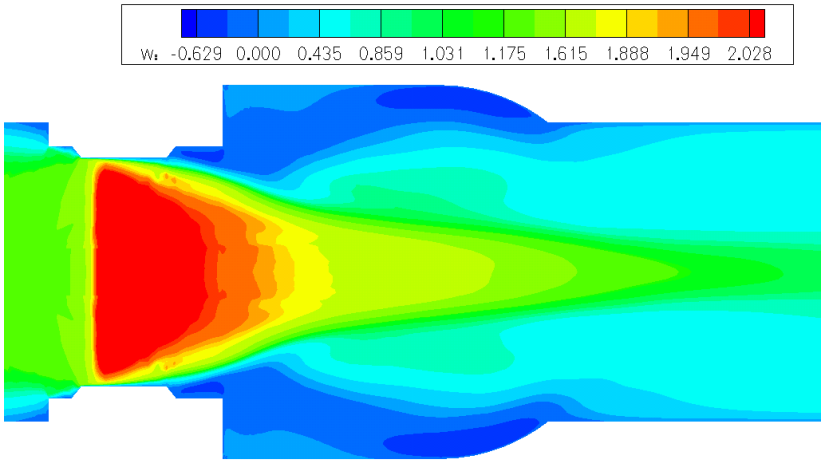


(b) $y = 0$ plane

Figure 71: Time averaged streamwise velocity contours (URANS)



(a) $x = 0$ plane

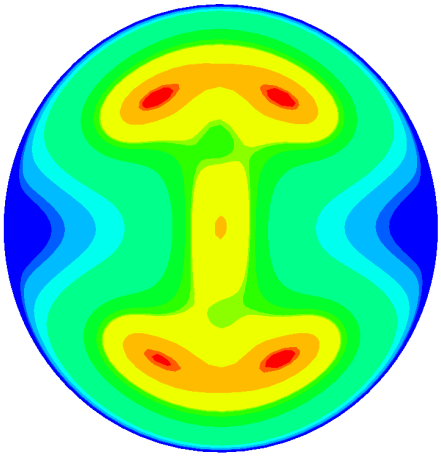


(b) $y = 0$ plane

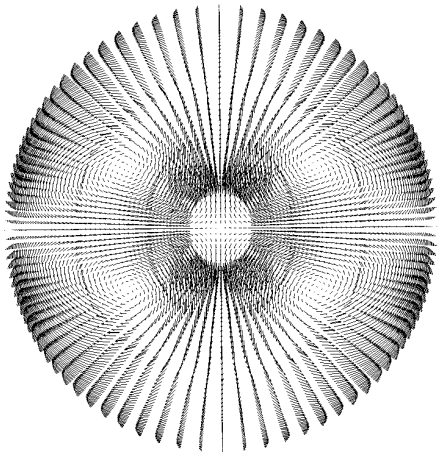
Figure 72: Time averaged streamwise velocity contours (DES)

flowfields undergo weak temporal oscillations about the mean. For the DES solution, on the other hand, the time-average picture is drastically different than the instantaneous images. The mean flow consists of only two pairs of streamwise vortices, whose cores are located approximately at the same positions as in the URANS flowfield. Yet the distribution of time-averaged streamwise momentum within the cross-section is substantially different between the two modeling approaches. Both solutions display the characteristic triple-jet structure but the URANS flowfield exhibits two fairly large pockets of low axial momentum in the regions where the common flow of the vortices transports the flow away from the wall along the horizontal symmetry plane. These pockets of low momentum are not nearly as pronounced in the DES flowfield. As a result the positive maxima of axial flow in the DES flowfield is much less than those in the URANS flowfield, a trend that is a direct consequence of mass conservation. In other words, DES yields a much more uniformly distributed axial momentum across the entire cross-section.

To validate the numerical simulations, we compare the time–averaged DES and URANS solutions with the experimental data in terms of profiles of mean streamwise velocity. The streamwise velocity profiles are compared at three different locations on three different planes parallel to the leaflets, which are shown in Fig. 75. The measurements were collected using the same PIV approach used for the laminar flow data and exhibit similar, albeit more pronounced, asymmetries as those discussed in the presentation of the laminar flow results. These asymmetries not withstanding, however, the comparisons are quite encouraging and are shown in Fig. 76. Overall, both the URANS and DES profiles are in reasonable agreement with the experimental observations. The location of the sinus recirculating region, the velocity peaks in the three jets, and, most importantly, the rapid decay of the triple-jet structure in the wake flow are all captured by the two methods with good accuracy. This latter feature of the flow is especially important since steady RANS models are well known to have great difficulty capturing the relaxation of a rapidly strained flow toward equilibrium. The fact that both approaches accurately resolve the rate of decay of the triple-jet structure suggests, therefore, that unsteady statistical models are quite promising as engineering predictive tools in complex flows. As previously discussed, an

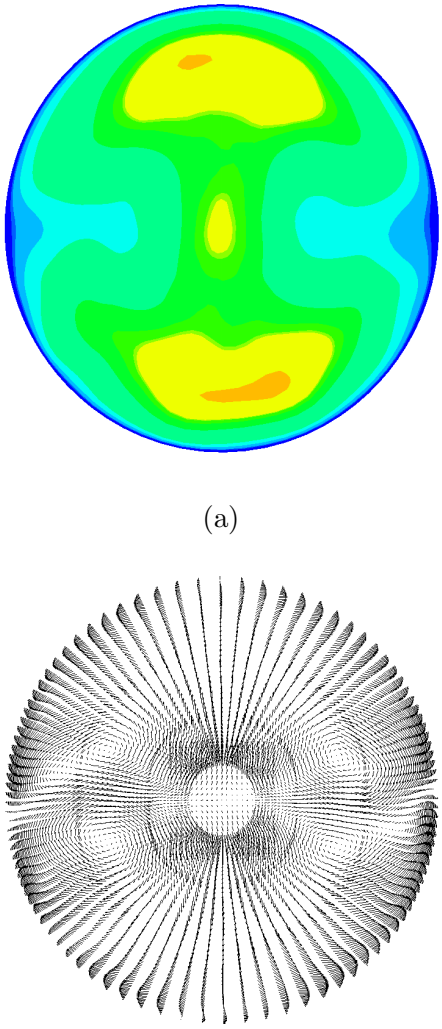


(a)



(b)

Figure 73: Cross sectional flow pattern of the time–averaged flow field (URANS)



(a)

(b)

Figure 74: Cross sectional flow pattern of the time–averaged flow field (DES)

important difference between the URANS and DES approach is the predictions within the sinus region. The DES yields a smaller region of reverse flow and this is in good agreement with the experimental measurements. In contrast, URANS consistently overpredicts the negative peak in the velocity profiles near the wall in almost all profiles. This difference, subtle as it may appear from the profiles shown in Fig. 76, can have a profound effect in the distribution of streamwise momentum within the cross-section as shown in Figs. 73 and 74. To understand the reasons for this significant discrepancy recall that URANS yielded essentially steady flow within the sinus region while DES resolved the shear layer instability, which excited complex unsteady modes within the sinus region. The multiple large-scale structures within the sinus resolved by DES appear, therefore, to play a major role in transporting and redistributing streamwise momentum and their resolution is critical prerequisite for accurate simulation of the recirculating flow patterns.

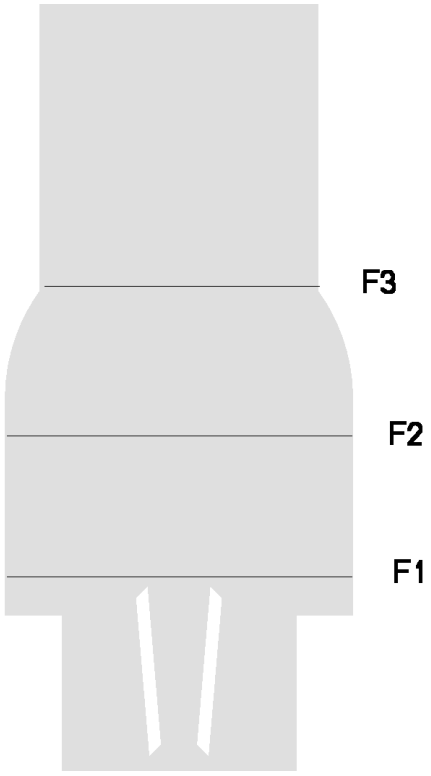
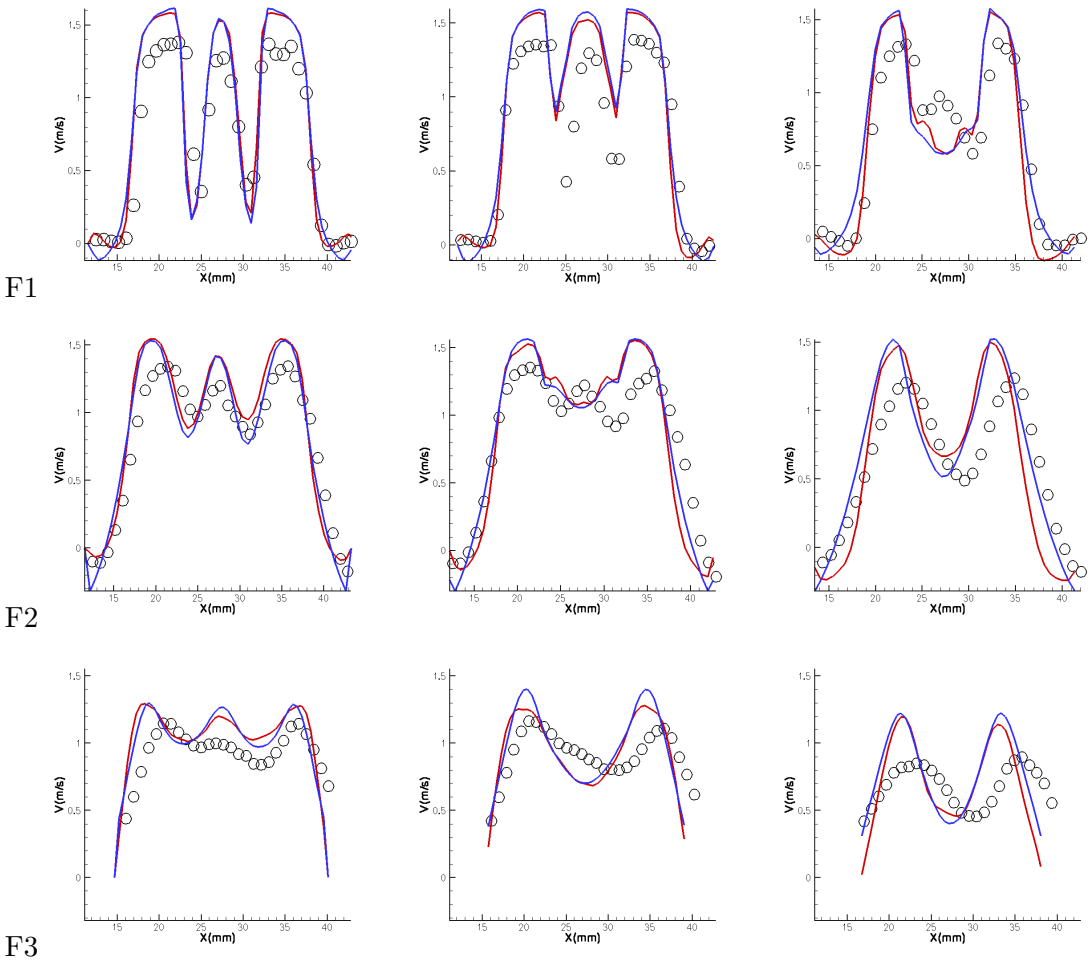


Figure 75: Locations of comparisons between numerical and experimental results



(a) Center (b) 0.3R (c) 0.6R

Figure 76: Comparisons between DES (red line), URANS (green line) and experimental measurements (circle) $Re = 6000$

4.2.2.3 Turbulence statistics

The flow rate at $Re = 6000$ is close to the peak systole flow rate of a circulation cycle. Blood flow at this condition is believed to experience fairly high levels of turbulent stresses, especially in the wake flow region. In Figs. 77 – 80 we show a sample of simulated turbulence statistics at the horizontal plane of symmetry. We include the primary turbulent shear stress $-\rho\overline{v'w'}$ in terms of three terms that contribute to it: (i) viscous stress, (ii) modeled stress, and (iii) directly resolved stress. We also show the resolved normal stress in the streamwise direction $-\rho\overline{w'w'}$. Note that we only show the streamwise normal stress because its magnitude was found to be three to four times greater than the magnitude of the other two normal stresses.

The viscous primary shear stress is defined as follows:

$$\tau_v = \mu(\frac{\partial V}{\partial z} + \frac{\partial W}{\partial y}) \tag{64}$$

where V, W are the y and z direction velocity components, respectively, and in a turbulent flow is typically expected to be very small relatively to the turbulent shear stress everywhere expect in the immediate vicinity of solid walls. As shown in Fig. 77, there are six zones of intense viscous shear on the $x = 0$ plane, which are generated either by the leaflets or the valve housing. As expected, the viscous shear stress magnitude reaches local maximum near the solid surfaces, be it the leaflet or the housing walls. The region of highest viscous stress appears to be the area close to the tips of the leeward side of leaflets. The DES solution peaks at a value of 433 $dyne/cm^2$ and the URANS solution peaks at a value of 450 $dyne/cm^2$. On the other side of the leaflets, the local maximum reaches at a value of 421 $dyne/cm^2$ for DES and 423$dyne/cm^2$ for URANS. Overall, the viscous shear stress from the two approaches are quite similar with the most significant differences occurring, as expected based on our earlier discussion, Sin the shear zone between the sinus and main channel flow.

Modeled and resolved turbulence shear stresses represent the portion of total turbulence shear stress modeled by the turbulence model and resolved directly by the unsteady calculation, respectively. The summation of these two variables shows the overall distribution

of the turbulence shear stress in the flow domain. Comparison of their relative magnitude, on the other hand, gives us some indication about how much energy is carried by the resolved large–scale coherent vortices versus the energy carried by the modeled small-scale turbulence. The modeled Reynolds stress $\left(-\rho\overline{v'w'}\right)_M$ is obtained as

$$\left(-\rho\overline{v'w'}\right)_M = \mu_t(\frac{\partial V}{\partial z} + \frac{\partial W}{\partial y}) \tag{65}$$

and its distribution is shown in Fig. 78. As seen in this figure, the turbulence shear stress modeled by both URANS and DES is primarily distributed along the sinus root shear layer. In both approaches, the modeled Reynolds stress reaches maximum near the end of the sinus root. However, the magnitude of this maximum value is quite different. The URANS solution peaks at a value near 460 $dyne/cm^2$ while the maximum value of DES solution is only about 110 $dyne/cm^2$. Furthermore, the pocket of maximum stress is much larger in the URANS than in the DES flowfields. This trend is consistent with the fact that URANS yields a steady flow in that region and, thus, all turbulent energy is accounted for by the model. The DES on the other-hand resolves a considerable portion of the energy in this region and, thus, yields smaller levels of modeled stress.

The resolved shear stress $\left(-\rho\overline{v'w'}\right)_R$ and $\left(-\rho\overline{w'w'}\right)_R$ are obtained by taking statistics of the instantaneous URANS and DES solutions . These statistics are calculated as

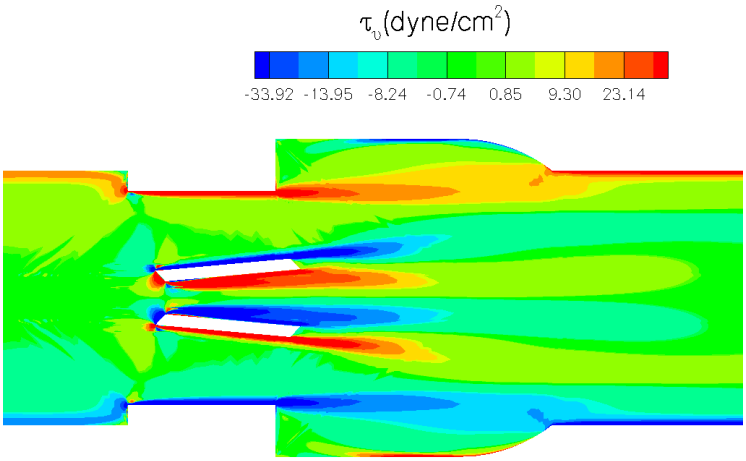
$$f(u_i'u_j') = -\rho\overline{u_i'u_j'} = \int_t (u_i - \overline{U_i})(u_j - \overline{U_j})dt \tag{66}$$

where u_i, j is the instantaneous solution obtained from the calculation and $\overline{U_i}, \overline{U_j}$ is time–averaging value of $u_{i,j}$. Fig. 79 shows the distribution of $\left(-\rho\overline{v'w'}\right)_R$. Both in URANS and DES solutions, the resolve shear stress dominates in the wake of leaflets, on each side of every leaflet. The resolved stress peaks at a value of 640 $dyne/cm^2$ for URANS and 610 $dyne/cm^2$ for DES. Same as observed in the modeled shear stress distribution, the difference between the URANS and DES results are located at the end of sinus region. The DES approach in this region clearly picks up the part of turbulence shear stress that was not modeled by the underlying turbulence model. This comparisons further shows that DES resolves directly a major part of turbulence shear stress in the sinus shear layer. As discussed before, the ability of DES to excite the unsteadiness of the large-scale flow in this

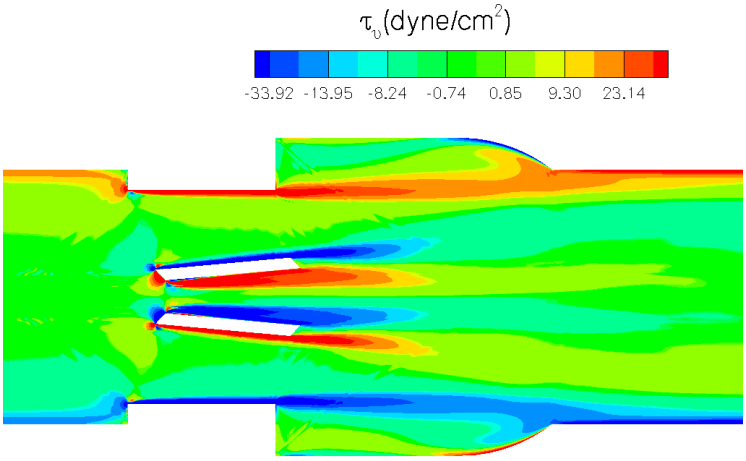
regions accounts for its successful prediction of the mean velocity profiles within the pocket of recirculating flow in the sinus.

The $-\rho\overline{w'w'}$ Reynolds stress is shown in Fig. 80. In the URANS solution, the normal Reynolds stress dominates in the main flow channel. The local maximum appears near the leeward side of each leaflet, at a value of 2265 $dyne/cm^2$. In the DES solution, the resolved normal stress is distributed across the entire flow channel. The area where the maximum $\left(-\rho\overline{w'w'}\right)_R$ is located is quite similar as the URANS result. The maximum value of $\left(-\rho\overline{w'w'}\right)_R$ in the DES solution reaches 2265 $dyne/cm^2$.

The results reported in this section suggest that (a) the turbulent stresses in the MHV flow are substantially larger than viscous stresses; (b) the maximum value of turbulence stresses is in the critical range and blood cells exposed in such stress levels are expected to experience lethal or sublethal damage; and (c) the regions close to the leading edge of the leaflets are the most dangerous area for blood cells, combining the elevated level of shear stress and foreign materials.

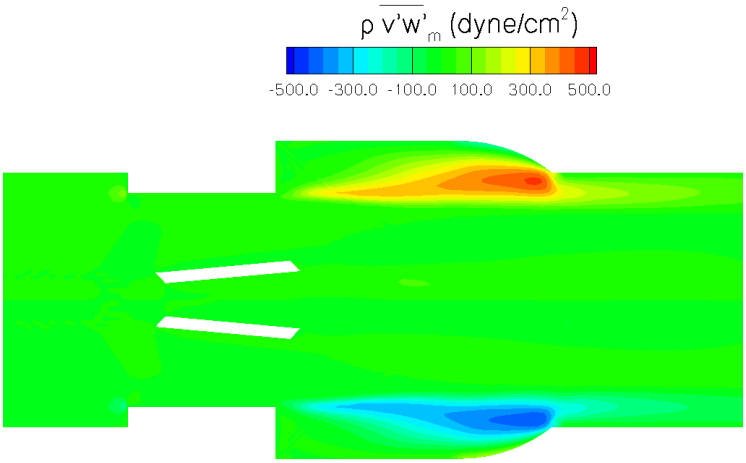


(a) URANS solution

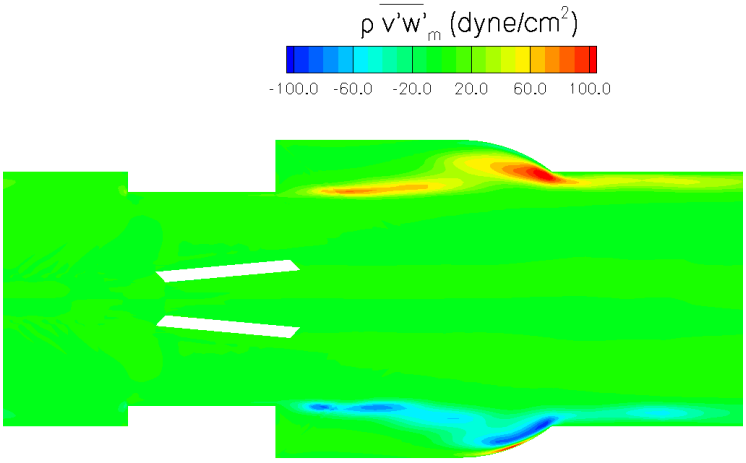


(b) DES solution

Figure 77: Viscous shear stress on $x = 0$ plane

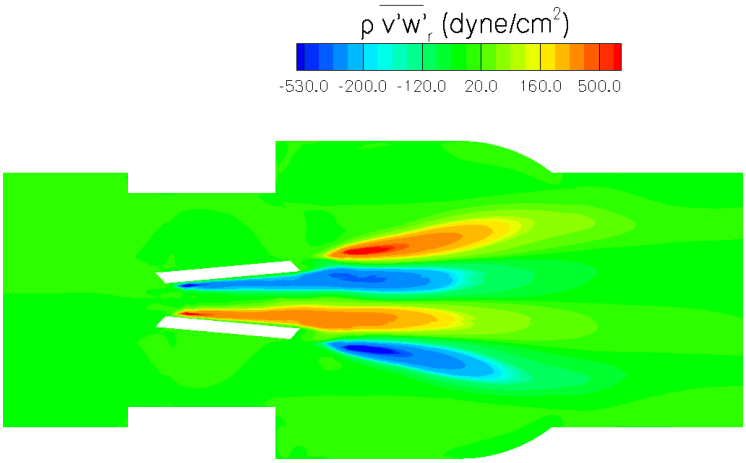


(a) URANS solution

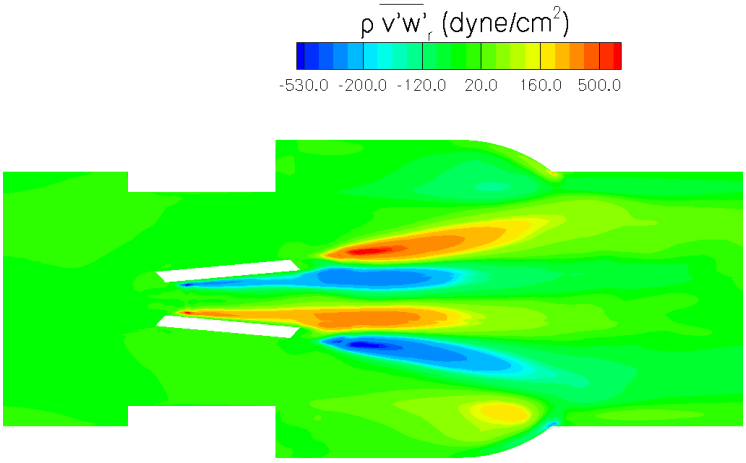


(b) DES solution

Figure 78: Modeled Reynolds stress on $x = 0$ plane

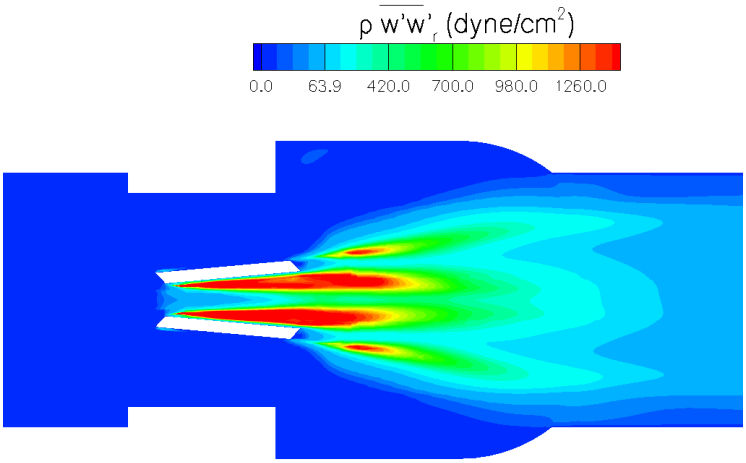


(a) URANS solution

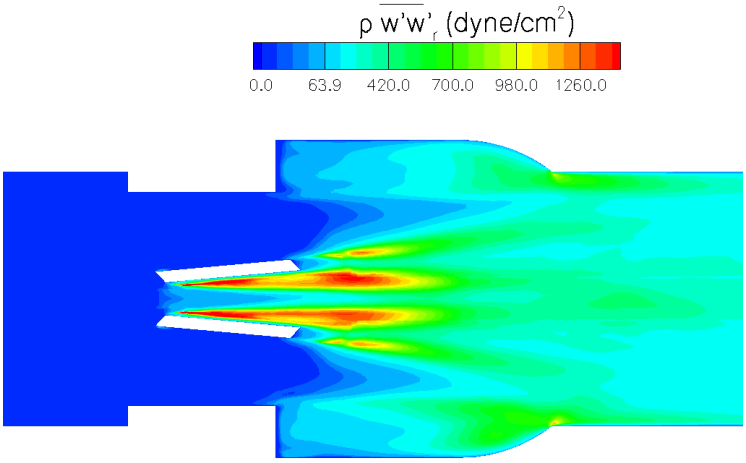


(b) DES solution

Figure 79: Resolved Reynolds stress $\rho\overline{v'w'}$ on $x = 0$ plane



(a) URANS solution



(b) DES solution

Figure 80: Resolved Reynolds stress $\rho\overline{w'w'}$ on $x = 0$ plane

CHAPTER V

CONCLUSIONS

This thesis sought: a) to develop an accurate, efficient, and versatile numerical solver that can accurately predict a broad range of complex engineering flows with unsteady statistical turbulence models; b) to validate the numerical solver through comparisons with experimental measurements; and c) to demonstrate the capabilities of the method by applying it to elucidate the complex physics of flows spanning a broad range of practical applications, from hydraulic engineering to bioengineering.

We developed a new state-of-the-art CFD solver capable of simulating a broad range of complex engineering flows at real-life Reynolds numbers. The method solves the three–dimensional incompressible unsteady Reynolds–averaged Navier–Stokes (URANS) equations closed with unsteady statistical turbulence models. Three such models are incorporated in the solver: the standard $k - \varepsilon$ model with wall functions, the Spalart–Allmaras model and the detached–eddy simulation (DES) model. The numerical solver employs domain decomposition with structured Chimera overset grids to handle complex, multi-connected geometries. The overset grid method divides a complex domain into a number of geometrically simpler, arbitrarily overlapping sub–domains, each of which can be easily discretized with an appropriate set of body-fitted, curvilinear coordinates. An iterative method is developed to solve the equations in the new composite domain. Solutions are first obtained in each sub–domain and information is transferred through the sub–domains interfaces by interpolation to specify boundary conditions at interfaces of adjacent subdomains.

The URANS equations are discretized in strong-conservation form using a finite volume method. The convective terms are discretized with three-point, second–order accurate, central differencing scheme plus third–order, fourth–difference matrix–valued artificial dissipation. The viscous terms are approximated with second–order central differencing. A

second-order accurate, dual- or pseudo-time stepping, artificial compressibility method is applied to integrate the discrete governing equations in time. During every physical time step, the equations are advanced in pseudo time with a block, implicit, approximate factorization scheme. The three different turbulent models are discretized as follows: the convective terms are discretized with flux–difference splitting upwind scheme; the rest of the spatial derivatives are approximated with second–order central difference. All turbulence equations are advanced in time using the same dual–time step method used for the URANS equations. To fully exploit the power of modern parallel computational facilities, the developed solver is parallelized using OpenMP.

The capabilities and versatility of the numerical method are demonstrated through the investigation of two widely different flow problems: a) flow through a geometrical complex array of bridge piers mounted both on a natural river reach and on a flat bed experimental flume; and b) flow in mechanical, bileaflet heart valve with the leaflets fixed in the fully-open position. In both cases, the complex geometries are successfully discretized with the Chimera overset grid method. Grid systems with millions of grid nodes are used and grid-refinement and other numerical dependency studies are carried out to explore the sensitivity of the computed solutions to various numerical parameters. For all simulated cases, large–scale unsteadiness appears naturally in our calculation as a result of excited mean-flow instabilities without posing explicit forcing on the flow. By advancing the time–dependent solution in a long time interval, we obtained statistically converged solutions. These solutions are compared with experimental data to validate our method. All comparisons show that by directly resolving the large–scale coherent vortices through the numerical procedure, the developed numerical solver can provide accurate predictions for real-life engineering flow problems with very complex geometries. In the following sections, we discuss the main observations and conclusions that follow from our work. Due to the different nature of the two flow problems studied herein, we will outline our conclusions in two separate sections: one dealing with the hydraulic engineering problem and the other with the cardiovascular flow problem.

5.1 Bridge foundation flows

1) The investigation of the flow around a single bent of piers mounted on the actual river bathymetry clearly showed that the numerical method can successfully capture the onset of naturally excited unsteady vortical pattern at the scale of the bridge foundation, while taking into account the large-scale bathymetry of the river reach within which the foundation is embedded. To the best of our knowledge our method is the first numerical technique to demonstrate this capability, which allows for accounting for the effect of the river bathymetry on the approach flow hydraulics in the vicinity of the foundation. Therefore, the method can serve as a powerful hydraulic engineering simulation tool as it can readily account for the multi-scale nature of most problems involving hydraulic structures embedded in natural river reaches.

2) The URANS equations even when closed with a relatively simple turbulence closure model, such as the standard $k - \varepsilon$ model used in this work, can capture the onset of large-scale unsteadiness within and in the wake of the complex bridge foundation. These coherent unsteady motions account for a significant percentage of the total turbulence kinetic energy in the wake of the foundation and their direct resolution via a URANS-type computation appears to be a critical prerequisite for quantitatively–accurate predictions of the mean flow and turbulence structure. The ability of the model to resolve essentially all major features of the flow in the wake of the foundation was especially encouraging as this is usually a region of the flow where standard steady RANS models are known to fail.

3) Comparisons between measured and computed streamwise mean velocity profiles at various locations upstream, within and downstream of the piers and at various flow depths show that the numerical model captures most experimental trends with good accuracy. Significant discrepancies between measurements and simulations were observed, however, in the distribution of the turbulence kinetic energy. The experiments reveal a highly asymmetric structure of the turbulence field with k being significantly higher on one side of the piers. Numerical tests with a slightly skewed approach flow provided strong evidence that the asymmetry of the turbulence structure observed in the experiment is due to the profound effect that even a very small misalignment between the approach flow and the axis

of the foundation has on the turbulence structure. Approach flow skewness is exacerbated by the complex geometry of the foundation and drastically alters the dynamics of the shear layers and large-scale vortex shedding from the two sides of the foundation. These findings suggest that unless the presence of asymmetries in the approach flow are either eliminated or precisely quantified in the experiment, it may be very difficult if not impossible to accurately resolve numerically the structure of turbulence in a specific experiment with a real-life bridge foundation such as the one considered herein.

4) Juxtaposition of the simulated flow patterns for the fixed, flat-bed case with the scour patterns obtained experimentally led to some interesting insights regarding the role of foundation-induced hydrodynamics on sediment transport and deposition processes. Even though the pocket of maximum scour depth was found to correlate well with the maximum shear velocity, our analysis suggests that the concept of critical bed shear stress alone may not be sufficient for modeling sediment transport processes in complex foundations. Regions of scour were found to also correlate with pockets of negative vertical velocity and the associated pockets of lateral divergence of limiting streamlines away from the obstacle. Regions of local deposition, on the other hand, were found to be linked to pockets of positive vertical velocity, which are accompanied by lateral convergence of the limiting streamlines toward the obstacle. Our results, therefore, point to the need for developing physics-based models of scour that account for the interaction between the foundation-induced hydrodynamics with the erodible bed. Such models should incorporate in their framework the highly three-dimensional nature of the mean flow and turbulence structure in the vicinity of the foundation and should also rely on a fully three-dimensional, unsteady hydrodynamic model such as the one we have developed and validated in this work.

5) The simulation of the flow past two pier bents further underscored the ability of the method to simulate complex geometries and underscored its potential to simulate flows past complete bridge sections. The computed results for this case showed that the introduction of the additional pier bent, which was placed at a considerable spanwise distance from the first bent, had little effect on the time–averaged flow velocity field. In spite of the relatively large spanwise spacing of the two bents, however, the presence of the second structure had

a profound effect on the distribution of the turbulence kinetic energy distribution in the vicinity of each set of piers. The effect was shown to be similar to that introduced by skewing the approach flow for the single bent case, namely the strengthening of the bent shear layer on one side of the bent and the appearance of higher levels of turbulence kinetic energy on that side.

6) The apparent sensitivity of the turbulence kinetic energy distribution around a bridge foundation to approach flow skewness and/or the presence of adjacent hydraulic structures could have significant implications insofar as the development of accurate scour prediction methodologies is concerned. There is a growing body of recent literature emphasizing the importance of stochastic modeling due to the turbulent nature of the flow in sediment transport calculations. Implementation of stochastic models of sediment transport in a hydrodynamic model requires the accurate description of the turbulence statistics near the foundation. The results we presented in this thesis clearly suggest, therefore, that quantitatively accurate predictions of scour under field conditions could necessitate calculations for the complete bridge foundation mounted on the actual river bed. Our method provides the first computational simulation tool that offers such capability.

5.2 Flow in mechanical heart valves

1) Time-accurate simulations were carried out for flow through a bileaflet mechanical heart valve mounted in a model axisymmetric aorta with the leaflets fixed in the fully open position and under steady (non-pulsatile) inflow conditions. The simulated Reynolds numbers covered a broad spectrum of physiological flow conditions, from the laminar to the fully turbulent regimes. The calculations were carried out on fine computational meshes, which were one order of magnitude finer than those used in numerical studies previously reported in the literature. Notwithstanding the simplifications we adopted (fixed leaflets, steady inflow), our simulations yielded the first ever fully three-dimensional and unsteady descriptions of heart valve flowfields and provided numerous novel insights into their complex hemodynamics.

2) For flows in the laminar regime, we found that at sufficiently low Reynolds numbers

the flow in a bileaflet MHV is steady and symmetric with respect the valve's geometric axes of symmetry. The flow in the wake of leaflets is dominated by two pairs of counterrotating streamwise vortices emanating from both sides of each leaflet near the valve housing.

3) For flows with Reynolds number larger than a certain threshold, the quadrant symmetry of the flow breaks and unsteady, coherent vortex shedding is observed in the wake of leaflets. For Reynolds numbers just above the threshold for the onset of unsteady, three-dimensional flow, unsteadiness first sets in the recirculating toroidal region within the aortic sinus as a result of the interaction between the slow moving fluid within the sinus and the leaflet shear layers. The streamwise vortex pairs in the wake of leaflets begin to undergo transverse oscillations and start to engage in a complex interaction with the near wall flow by extracting vorticity of opposite sign from the wall boundary layers. As the Reynolds number increases, this interaction leads to the formation of new vortices whose mutual interactions give rise to a very complex, highly unsteady and seemingly chaotic wake flow.

4) Spectral analysis of time series in the wake region show that the onset of unsteadiness is not characterized by the emergence of one simple, periodic mode. Instead two incommensurate frequencies appear in the flow, which give rise to a very complex, quasi-periodic dynamics.

5) An important phenomenon that is clearly visible in all simulated laminar flowfields is the apparent switching of the axes of the main orifice jet downstream of the leaflets. This jet forms between the two leaflets and evolves under the action of the two vortex pairs on each side of the aorta. The common flow of these pairs is directed away from the wall toward the center of the cross-section and, thus, acts to compress the main orifice jet laterally. This lateral straining causes the jet to switch its major and minor axes few aorta diameters downstream of the leaflets. The phenomenon of axis switching is well documented in the literature of non-circular free jets. In such flows, the jet axes switch as a result of the interaction of the jet with its own self-induced vorticity. To the best of our knowledge, however, this is the first time that this complex phenomenon is documented in a bileaflet heart valve flow.

6) Turbulent flow simulations were carried out with URANS and DES. Both models yielded unsteady solutions with rich coherent vortex shedding. However, the instantaneous solutions are quite different in the sense of vortex dynamics. In the URANS solution, only the largest scale vortices are directly resolved and all small scale motions are smoothed out. The instantaneous solution is found to undergo small amplitude unsteady oscillations around the time–averaged solution and, thus, most turbulent energy is modeled by the turbulence model. Within the sinus region, URANS predicts a single, quasi-steady toroidal vortex and the shear layer between the recirculating sinus flow and the leaflet minor orifice jets remain stable. On the other hand, the DES solution, which is a hybrid modeling framework combining URANS and LES, yields a very complicated, dynamically rich vortical flow in the entire channel. Small scale vortex shedding was obtained in the wake of the leaflets and the multiple vortical structures were found in the recirculation region within the sinus. These vortices are very energetic and exhibit rich, albeit low frequency, dynamics continuously merging with each other to form larger vortices and then splitting again, and so on.

7) The simulations were validated through comparisons with experimental data. In both the laminar and turbulent flow regime simulations, the obtained numerical results capture most experimental trends with reasonable accuracy. In the laminar flow regime, the direct numerical simulation reproduced the three-dimensional distribution of the momentum and the rate at which the three jets diffuse and merge in the wake flow region with very good accuracy.

8) For the turbulent flow simulation, both URANS and DES methods predict the jet structure and the much faster compared to the laminar case rate of decay of the three jets reasonably well. DES though leads to consistently better predictions of the streamwise velocity within the recirculating flow region in the sinus. In this region, URANS tends to consistently overpredict the magnitude of reversed flow. This finding suggests that the large-scale, low-frequency eddy motions resolved by DES in the sinus region are responsible for most of the transport and redistribution of streamwise momentum and needs to be resolved directly for accurate prediction of the mean flow. As discussed above, URANS yields a quasi-steady flow in the sinus and consequently fails to capture the dynamics in

this very important region of the flow. Note that this is a region where blood elements could potentially be trapped for long time and begin to aggregate to form thrombi. Thus, the accurate resolution of the flow in the aortic sinus is very important if the computed flowfields are to be used to examine the presumed link between valve hemodynamics and clinical complications.

9) The numerical simulations provided novel insights into the complex hemodynamics of the valve especially in regions which are not accessible by experimental methods. Instantaneous limiting streamlines and contours of wall shear stress magnitude along the sinus wall revealed a very complex, and highly unsteady near-wall velocity field with multiple vortices and regions of high shear stress levels. Being able to predict such complex flow patterns is critical if the numerical flowfields are to be used to assess the potential effect of the near wall flow on the bio-chemistry of cells that line the endothelium. Furthermore, the levels of turbulence stresses downstream of the valve at the near peak-systole Reynolds number were shown to be comparable to stresses measured in laboratory experiments. The numerical flow fields, however, further yield the full three-dimensional distribution of the turbulent stresses and can thus be used to identify regions where lethal or sub–lethal damage of blood cells is likely to occur. All these insights show the great potential of CFD as a powerful analysis and design tool in cardiovascular flow investigations.

Finally, it is important to point out that both the hydraulic engineering and the cardiovascular flow problems we studied in this work underscore the need for a close synergy between laboratory experiments and numerical simulations. The former can provide data to validate the latter. The numerical simulations can in turn guide the experimental investigation by identifying regions where important flow physics occur and further suggesting the spatial and temporal experimental resolution needed to access these complex physics. Therefore, the numerical simulations can be used to facilitate the collection of better quality and more relevant experimental data, which in turn can be used for a meaningful validation of the predictive capabilities of the numerical tools.

5.3 Remarks for future work

In the current work, the flows around bridge foundations are modeled with rather simple $k-\varepsilon$ model with wall functions. Although great success has been obtained with such a simple turbulence model, the capabilities of the developed numerical model can be further improved by implementing more advanced turbulence models. Hybrid approaches such as the DES may have great potential in such flows and deserve further investigation.

A sediment transport model needs to be developed and implemented into the hydrodynamic model, which can simulate the deformation of the erodible bed due to the foundation-induced hydrodynamics in a strongly coupled manner. As we have clearly demonstrated in this work, however, for both undertakings to be successful high-resolution laboratory experiments must be tightly integrated with the numerical model development efforts. Numerical simulations can help guide the design of the laboratory experiments, which in turn will yield the high-resolution data needed for physics-based numerical modeling of the complex hydrodynamic and scouring processes at real-life bridge foundations.

The investigation of heart valve flow in this work focused on steady inflow conditions, while the actual physiological flow is pulsatile and the leaflets are free to move. The pulsatile inflow condition will drastically change the flow physics and vortical dynamics in MHV flows and, thus, pulsatile simulations with moving leaflets are essential for obtaining hemodynamical relevant results. The computational expense for such flow investigations, however, is anticipated to be much higher than that of the current steady inflow studies. Therefore, the efficiency of the iterative solver used in this work needs to be improved further. A variety of fast iterative solvers for solving the pseudo-compressible form of the Navier-Stokes equations, such block relaxation schemes with LU decomposition, need to be implemented and investigated.

The numerical method needs to also be extended to account for the motion of the leaflets and the ensuing fluid/structure interaction. The current version of the code has already built in it the capability to account for the motion of the leaflets but such simulations have not been attempted so far. Overset grids could very well be shown adequate for simulating the moving leaflets but numerical difficulties are anticipated with this approach

when the valve closes. We thus propose that the overset grid approach should be combined with the recently developed hybrid Cartesian/Immersed Boundary approach of Gilmanov et al. (2003). Overset body-fitted grids could be used to discretize the aorta within which a fine mesh subdomain containing the leaflets is embedded. Within this subdomain, the moving leaflets are tracked as sharp interfaces using the approach of Gilmanov et al. (2003). Such hybrid overset/Cartesian mesh approach will yield an extremely powerful numerical methodology that can be used to simulate the flow through a bileaflet valve embedded in an anatomically realistic aorta geometry.

During the cardiac cycle the Reynolds number ranges from 0$\sim$7000 and, thus, the flow in MHV undergoes periodic laminar to turbulence transition and relaminazation. In this work we have demonstrated the potential of DES for predicting fully turbulent flow near peak systole. The performance of this approach, however, needs to be further investigated in the transitional flow regime. Such undertaking will also require very detailed experimental data under physiological, pulsatile flow conditions to guide the refinement and validation of a suitable turbulence model for transitional flows.

APPENDIX A

MODAL MATRICES

The modal matrix and its reverse in Eq. 51 are

$$M_j = \frac{1}{\sqrt{g^{jj}}} \begin{bmatrix} -\beta C_j & \beta C_j & 0 & 0 \\ \beta\xi^j_{x_1} + u_1\lambda^j_1 & \beta\xi^j_{x_1} + u_1\lambda^j_2 & -\xi^j_{x_2} & -\xi^j_{x_3} \\ \beta\xi^j_{x_2} + u_2\lambda^j_1 & \beta\xi^j_{x_2} + u_2\lambda^j_2 & \xi^j_{x_1} + \xi^j_{x_3} & -\xi^j_{x_3} \\ \beta\xi^j_{x_3} + u_3\lambda^j_1 & \beta\xi^j_{x_3} + u_3\lambda^j_2 & -\xi^j_{x_2} & \xi^j_{x_1} - \xi^j_{x_2} \end{bmatrix}$$

$$M_j^{-1} = \frac{\sqrt{g^{jj}}}{2C_j^2} \begin{bmatrix} -\frac{\lambda^j_2}{\beta} & \xi^j_{x_1} & \xi^j_{x_2} & \xi^j_{x_3} \\ -\frac{\lambda^j_1}{\beta} & \xi^j_{x_1} & \xi^j_{x_2} & \xi^j_{x_3} \\ -\frac{1}{\beta}\left(\lambda^j_2 d^j_1 + \lambda^j_1 d^j_2\right) & -\left(\frac{2C_j^2}{k^j} + \xi^j_{x_1} d^j_5\right) & \frac{2C_j^2}{k^j} - \xi^j_{x_2} d^j_5 & -\xi^j_{x_3} d^j_5 \\ -\frac{1}{\beta}\left(\lambda^j_2 d^j_3 + \lambda^j_1 d^j_4\right) & -\left(\frac{2C_j^2}{k^j} + \xi^j_{x_1} d^j_6\right) & -\xi^j_{x_2} d^j_6 & \frac{2C_j^2}{k^j} - \xi^j_{x_3} d^j_6 \end{bmatrix}$$

where

$$C_j = \sqrt{\left(U^j\right)^2 + \beta g^{jj}}$$

and

$$\lambda^j_1 = U^j - C_j, \quad \lambda^j_2 = U^j + C_j, \quad \lambda_3 = \lambda_4 = U^j$$

$$\begin{aligned} k^j &= \xi^j_{x_1} + \xi^j_{x_2} + \xi^j_{x_3} \\ d^j_1 &= \tfrac{1}{k^j}\left[\beta\left(\xi^j_{x_2} - \xi^j_{x_1}\right) + \lambda^j_1\left(u_2 - u_1\right)\right] \\ d^j_2 &= \tfrac{1}{k^j}\left[\beta\left(\xi^j_{x_2} - \xi^j_{x_1}\right) + \lambda^j_2\left(u_2 - u_1\right)\right] \\ d^j_3 &= \tfrac{1}{k^j}\left[\beta\left(\xi^j_{x_3} - \xi^j_{x_1}\right) + \lambda^j_1\left(u_3 - u_1\right)\right] \\ d^j_4 &= \tfrac{1}{k^j}\left[\beta\left(\xi^j_{x_3} - \xi^j_{x_1}\right) + \lambda^j_2\left(u_3 - u_1\right)\right] \\ d^j_5 &= \tfrac{2}{k^j}\left[\beta\left(\xi^j_{x_2} - \xi^j_{x_1}\right) + \lambda^j_3\left(u_2 - u_1\right)\right] \\ d^j_6 &= \tfrac{2}{k^j}\left[\beta\left(\xi^j_{x_3} - \xi^j_{x_1}\right) + \lambda^j_4\left(u_3 - u_1\right)\right] \end{aligned}$$

The absolute value of A^j can be expressed as:

$$|A^j| = \frac{1}{2JC_j^2} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

where

$$\begin{aligned}
a_{11} &= C_j \left(|\lambda_2^j| \lambda_1^j - |\lambda_1^j| \lambda_2^j \right) \\
a_{12} &= \beta \xi_{x_1}^j C_j \left(|\lambda_1^j| - |\lambda_2^j| \right) \\
a_{13} &= \beta \xi_{x_2}^j C_j \left(|\lambda_1^j| - |\lambda_2^j| \right) \\
a_{14} &= \beta \xi_{x_3}^j C_j \left(|\lambda_1^j| - |\lambda_2^j| \right) \\
a_{21} &= |\lambda_1^j| \left(u_1 g^{jj} - \xi_{x_1}^j \lambda_2^j \right) + |\lambda_2^j| \left(u_1 g^{jj} - \xi_{x_1}^j \lambda_1^j \right) - 2|\lambda_3^j| \left(u_1 g^{jj} - \xi_{x_1}^j U^j \right) \\
a_{22} &= |\lambda_1^j| \left[u_1 \xi_{x_1}^j \lambda_1^j + \beta (\xi_{x_1}^j)^2 \right] + |\lambda_2^j| \left[u_1 \xi_{x_1}^j \lambda_2^j + \beta (\xi_{x_1}^j)^2 \right] \\
&\quad + 2|\lambda_3^j| \left[(U^j)^2 - \xi_{x_1}^j u_1 U^j + \beta (\xi_{x_2}^j)^2 + \beta (\xi_{x_3}^j)^2 \right] \\
a_{23} &= \xi_{x_2}^j \left[|\lambda_1^j| \left(u_1 \lambda_1^j + \beta \xi_{x_1}^j \right) + |\lambda_2^j| \left(u_1 \lambda_2^j + \beta \xi_{x_1}^j \right) - 2|\lambda_3^j| \left(u_1 U^j + \beta \xi_{x_1}^j \right) \right] \\
a_{24} &= \xi_{x_3}^j \left[|\lambda_1^j| \left(u_1 \lambda_1^j + \beta \xi_{x_1}^j \right) + |\lambda_2^j| \left(u_1 \lambda_2^j + \beta \xi_{x_1}^j \right) - 2|\lambda_3^j| \left(u_1 U^j + \beta \xi_{x_1}^j \right) \right] \\
a_{31} &= |\lambda_1^j| \left(u_2 g^{jj} - \xi_{x_2}^j \lambda_2^j \right) + |\lambda_2^j| \left(u_2 g^{jj} - \xi_{x_2}^j \lambda_1^j \right) - 2|\lambda_3^j| \left(u_2 g^{jj} - \xi_{x_2}^j U^j \right) \\
a_{32} &= \xi_{x_1}^j \left[|\lambda_1^j| \left(u_2 \lambda_1^j + \beta \xi_{x_2}^j \right) + |\lambda_2^j| \left(u_2 \lambda_2^j + \beta \xi_{x_2}^j \right) - 2|\lambda_3^j| \left(u_2 U^j + \beta \xi_{x_2}^j \right) \right] \\
a_{33} &= |\lambda_1^j| \left[u_2 \xi_{x_2}^j \lambda_1^j + \beta (\xi_{x_2}^j)^2 \right] + |\lambda_2^j| \left[u_2 \xi_{x_2}^j \lambda_2^j + \beta (\xi_{x_2}^j)^2 \right] \\
&\quad + 2|\lambda_3^j| \left[(U^j)^2 - \xi_{x_2}^j u_2 U^j + \beta (\xi_{x_2}^j)^2 + \beta (\xi_{x_3}^j)^2 \right] \\
a_{34} &= \xi_{x_3}^j \left[|\lambda_1^j| \left(u_2 \lambda_1^j + \beta \xi_{x_2}^j \right) + |\lambda_2^j| \left(u_2 \lambda_2^j + \beta \xi_{x_2}^j \right) - 2|\lambda_3^j| \left(u_2 U^j + \beta \xi_{x_2}^j \right) \right] \\
a_{41} &= |\lambda_1^j| \left(u_3 g^{jj} - \xi_{x_3}^j \lambda_2^j \right) + |\lambda_2^j| \left(u_3 g^{jj} - \xi_{x_3}^j \lambda_1^j \right) - 2|\lambda_3^j| \left(u_3 g^{jj} - \xi_{x_3}^j U^j \right) \\
a_{42} &= \xi_{x_1}^j \left[|\lambda_1^j| \left(u_3 \lambda_1^j + \beta \xi_{x_3}^j \right) + |\lambda_2^j| \left(u_3 \lambda_2^j + \beta \xi_{x_3}^j \right) - 2|\lambda_3^j| \left(u_3 U^j + \beta \xi_{x_3}^j \right) \right] \\
a_{43} &= \xi_{x_2}^j \left[|\lambda_1^j| \left(u_3 \lambda_1^j + \beta \xi_{x_3}^j \right) + |\lambda_2^j| \left(u_3 \lambda_2^j + \beta \xi_{x_3}^j \right) - 2|\lambda_3^j| \left(u_3 U^j + \beta \xi_{x_3}^j \right) \right] \\
a_{44} &= |\lambda_1^j| \left[u_3 \xi_{x_3}^j \lambda_1^j + \beta (\xi_{x_3}^j)^2 \right] + |\lambda_2^j| \left[u_3 \xi_{x_3}^j \lambda_2^j + \beta (\xi_{x_3}^j)^2 \right] \\
&\quad + 2|\lambda_3^j| \left[(U^j)^2 - \xi_{x_3}^j u_3 U^j + \beta (\xi_{x_1}^j)^2 + \beta (\xi_{x_2}^j)^2 \right]
\end{aligned}$$

APPENDIX B

NUMERICAL SOLVER

B.1 Introduction

A numerical software package was developed, based on the algorithm described in Chapter 2. The software package includes a pre-processor, a kernel solver and a post-processor. Generally, all these three parts are needed to run a simulation on multi-block grid systems. The pre-processor handles the grid-connectivity tasks, with the grid connectivity information on the output. The kernel solver reads in the grid and grid connectivity information and obtain solutions to the flow. In order to save space, all results obtained from the kernel solver are saved in unformatted format and the post-processor can be used to transfer these files into the format that tecplot can understand. All codes are written in Fortran 90 and many new features, such as Modules, dynamic allocatable arrays, partial reference of arrays, etc., are used.

B.2 Pre-processor

The pre-processor is designed to provide necessary grid connectivity information for the kernel solver. Since for problems where the boundary geometries are fixed, the relative positions between the multi–blocks don't change and thus it makes perfect sense to only find the grid connectivity information once and before the kernel solver is started. The information can be stored in a file and be reused throughout the simulation. Basically what the pre-processor does is to find, for each node located on the artificial interfaces resulted from the domain decomposition procedure, the cell that encompasses the boundary node and calculate the tri-linear interpolation coefficients based on the relative geometry location. The detail algorithm can be found in Tang (2001).

The source code of the pre-process is named donor_cell.f90. It requires two input files: grid.dat and bcs.dat. File "grid.dat" stores the grid geometry information and "bcs.dat"

provides each sub–domain's boundary conditions. The file "grid.dat" is organized as follows (written in Fortran 90):

```
open(1, file = "grid.dat", format  = "unformatted")
do nn = 1, nzone
    write(1) nz
    write(1) im(nz), jm(nz), km(nz)
    write(1) (((x(i,j,k,nz),i=1,im(nz)),j=1,jm(nz)),k=1,km(nz))
    write(1) (((y(i,j,k,nz),i=1,im(nz)),j=1,jm(nz)),k=1,km(nz))
    write(1) (((z(i,j,k,nz),i=1,im(nz)),j=1,jm(nz)),k=1,km(nz))
end do
```

In the above code, nzone is the total number of sub-domains. As can be seen from the above code, the grid file is organized in a *domain-to-domain* manner, with each sub-domain starts from it's numbering (first block, second block, etc.). The line following the number is the dimensions of the sub-domain, in *i,j, and k* directions respectively. It's followed by the detail *x,y, and z* coordinates information, with each coordinate in a line. Multi-block overset grid can be generated with in-house developed code, or by using commercial software, such as Gridgen from Pointwise.com.

"bcs.dat" provides the pre-processor the boundary conditions. The format reads as:

```
open(2, file = "bcs.dat")
read(2,*) nzone

! out boundary condition
do nz=1,nzone
    read(2,*) nint(nz), (zint(nz,m), m = 1, nint(nz))
    read(2,*) (bctype(iface, nz), iface = 1, 6)
end do

! inner boundary condition for grid blanking
```

```
do nz=1,nzone
    read(1,*) blank_type(nz)
    read(1,*) nblank(nz),(zblank(nz,m),m=1,nblank(nz))

    if(blank_type(nz) == 1) then
          read(1,*) (blktype(iface,nz),iface=1,6)
          read(1,*) isblk(nz),ieblk(nz),jsblk(nz),jeblk(nz),ksblk(nz),keblk(nz)
    else if (blank_type(nz) == 2) then
          print '(/a5,i2,a/)', 'Zone ', nz, ' uses general blanking'
    else if (blank_type(nz) == 0) then
          print '(/a5,i2,a/)', 'Zone ', nz, ' uses no blanking'
    end if
 end do
```

Array nint(nz) means numbers of interfaced sub-domains for sub-domain nz, and zint(nz,m) stores which sub-domains are interfacing with sub-domain nz. The first part of boundary conditions are the outer boundary. Since we are using structured-grid for each sub-domain discretization, there are six boundaries for each sub-domain, $i_{min}, i_{max}, j_{min}, j_{max}, k_{min}\ and\ k_{max}$ respectively. "bctype(iface,nz)" stores corresponding boundary conditions for these six boundaries, with a number for each boundary. If bctype(iface,nz) == 0, then that specific boundary is an interface boundary and needs to be searched for grid connectivity information on that boundary. The inner boundary condition is designed for cases that need to blank part of the grid. For example, in the case of flow around bridge pier mounted on actual riverbed, whose grid is shown in Fig. 8, both block 1 and block 2 contains the whole geometry of bridge piers and the flow information in the vicinity of the piers is provided with the subdomains encompassing these piers (blocks 3–6). The areas around these piers in blocks 1 and 2 needs to be blocked out in the calculation. There are two types of blanking designed in the solver: specific blanking and general blanking. The specific blanking is for cases as shown in Fig. 8, where the blanked boundaries has a common

(or fixed) surfaces where blanking can be applied (for example, the thick black lines located in the middle of block 2 in Fig. 8). General blanking is designed for cases that such a common blanking surfaces don't exit (which is very useful for very complex geometries, such as the bileaflet flow calculation). If blank_type(nz) == 1, then the specific blanking is used. If blank_type(nz) == 2, general blanking applies. Other numbers of blank_type(nz) means no blanking. "nblank(nz)" tells the pre-processor how many sub-domains the searching sub-domain contains and zblank(nz,m) is the numbering of these contained sub-domains. "blktype(iface,nz)" provides the similar boundary conditions as bc-type(iface,nz) and $isblk(nz), ieblk(nz), jsblk(nz), jeblk(nz), ksblk(nz), keblk(nz)$ store the exact positions where the blanking starts and ends in $i, j, andk$ directions.

The solution from the pre-processor is stored in "interface.dat", which is an unformatted file. The kernel solver directly reads grid.dat and interface.dat as inputs.

B.3 Kernel solver

The kernel solver is the implementation of the numerical methods described in Chapter 2. Its function is to read in the geometry and control information, calculation the unsteady flow field and write the results into a series of files. The solver consists of the following parts: Modules, I/O part, URANS equations solver, turbulence equations solver, boundary interpolations and boundary conditions. Modules, stored in the file "MODULES.f90", is where all variables are defined. Three different modules are defined in this file: global_variables, local_variables and geometries. Dynamic allocatable arrays, which is supported in Fortran 90, is widely used in this solver. All large arrays are defined as allocatable and their actual size is defined upon execution, thus achieving the most efficient usage of available memory. The definition of the most important variables are provided in Table 2.

The flow chart of the kernel solver is shown in Fig. 81. Basically, it begins with the input of geometry information and control data, makes a initial guess of the flow field (by the user or from previous calculations), and then advances the URANS equations on each sub-domain for a single pseudo-time step. If it's a turbulent flow simulation, then the turbulence equations are also advanced in pseudo-time in each sub-domain. After all

Table 2: Important variables defined in MODULES.f90

qnew_global	:	$(p, u, v, w)^T$ at new pseudo time step *l+1*
qold_global	:	$(p, u, v, w)^T$ at old pseudo time step l
csi_global	:	ξ_{x_i}
eta_global	:	η_{x_i}
zet_global	:	ζ_{x_i}
aj_global	:	Jacobian
rh_global	:	Right hand side terms
wd_global	:	Wall distances to six surfaces
dt_global	:	local time step for URANS equations
dtkw_global	:	local time step for turbulence equations
qreal_global	:	solutions at time step n
qrealm1_global	:	solutions at time step *n-1*
qt_global	:	Turbulence terms at new pseudo time step *l+1*
qtold_global	:	Turbulence terms at old pseudo time step l
qtreal_global	:	Turbulence terms at time step n
qtrealm1_global	:	Turbulence terms at time step *n-1*
x_, y_, and z_global	:	coordinates of grid nodes
img, jmg, kmg	:	dimensions in i, j, and k directions
ln	:	connection between a multi–dimension array and a single dimension array
kgs_g, kgs	:	boundary condition types

equations are advanced in a pseudo-time step, the variables on the interface nodes are updated through interpolation, with the grid connectivity information obtained from the pre-processor. Boundary conditions are applied on all physical boundaries after the interpolation. Finally the residuals of the pseudo-time integration are calculated, by comparing the differences between two pseudo-time step solutions. If the residuals are smaller enough then the obtained pseudo-time solutions are considered as the physical time step solution; otherwise, repeat the above pseudo-time iteration. Typically, if residuals drop two-three orders, it can be considered as converged and this typically requires 15-25 pseudo-time iterations for ADI scheme. The most important subroutines and their functions are listed in Table 3.

The developed code is written in Fortran 90 and can be compiled with almost all available F90 compilers. We use gnumake to organize the file compilation and detail usage of gnumake can be found on http://www.gnumake.org. File "MODULES.f90" has to be compiled with "-c" option of F90 compiler before we can "make" the final executable. In this calculation, several different compilers are tested: Intel F90 compiler, NAG Fortran 95 compiler, and SGI Fortran 90 compiler. To obtain best performance from these different compilers, the following compiler options are suggested:

Intel F90 compiler: ifort -O3 -axP -tpp7. OpenMP support can be activated with "-openmp".

NAG Fortran 95 compiler: f95 -O4

SGI Fortran 90 compiler: f90 -n32 -mips4 -Ofast=IP27. OpenMP support can be activated with "-mp".

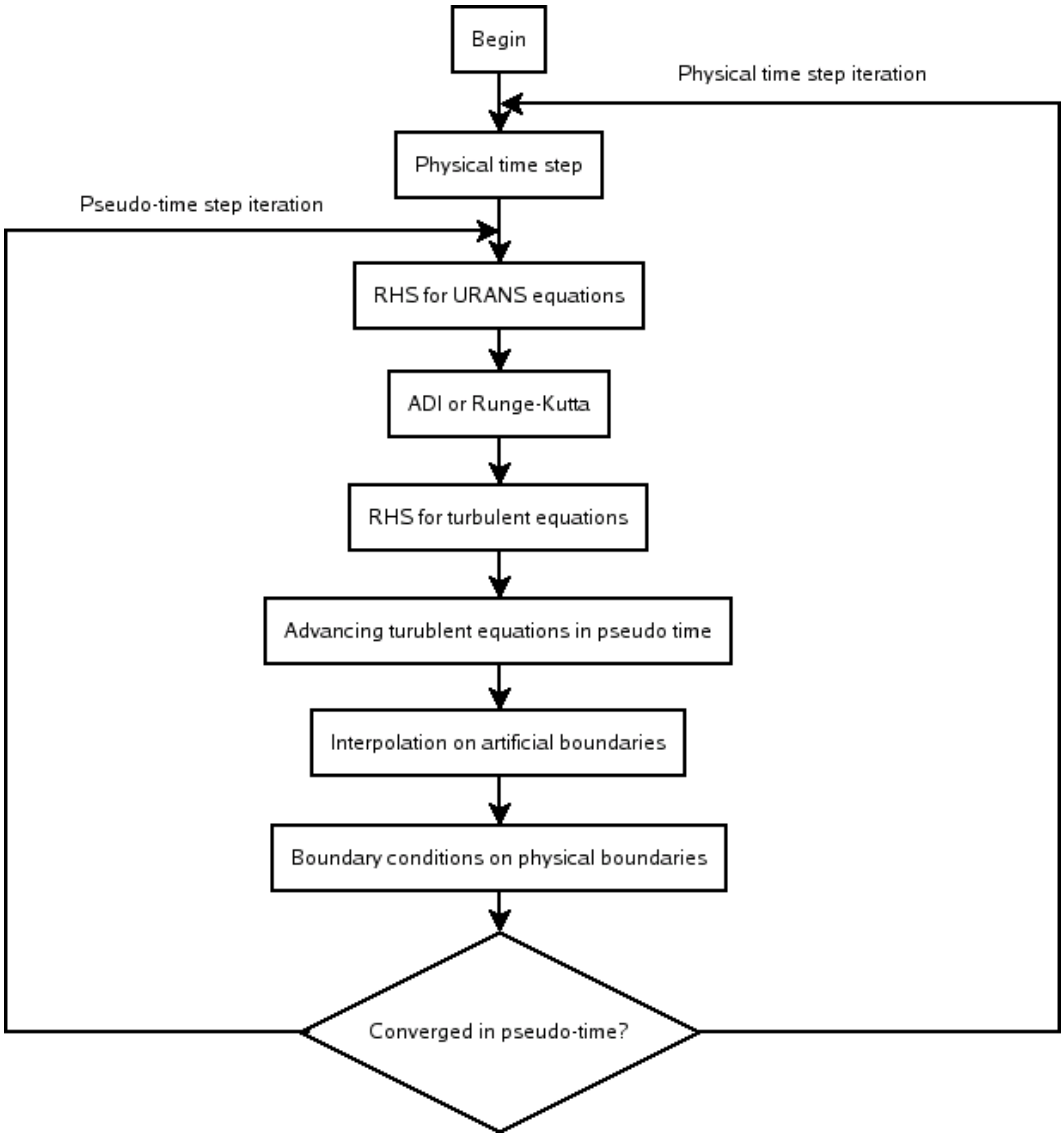


Figure 81: Flow chart of kernel solver

Table 3: Subroutines

metrics	:	calculate the metrics of the geometric transformation
contra	:	calculate contravariant velocity components
spectral	:	calculate local pseudo-time step
adiss4m	:	4th-order scalar dissipation
adiss2m	:	2nd-order scalar dissipation
adismatrix	:	Matrix-valued dissipation
visc_m1	:	viscous dissipation of URANS equations
convec	:	convective terms of URANS equations
unstead	:	unsteady time derivatives
rgku	:	Runge-Kutta scheme
rsmooth	:	residual smoothing (only used for Runge-Kutta)
rhs	:	calculate right hand side terms for URANS equations
fluxsp_m	:	flux splitting upwinding scheme
wall_functions	:	wall functions for $k - \varepsilon$ model
new_t	:	Newton-Rhapson non-linear equation solver
sgtsv	:	tridiagonal equation solver from LAPACK
adi_full	:	Beam and Warming approximate factorization scheme
adi_full_b	:	specific blanking
adi_full_gb	:	general blanking
solver	:	main solver for URANS equations
solver_b	:	specific blanking
solver_gb	:	general blanking
adi_t	:	turbulence equations solver for $k - \varepsilon$ model
adi_t_b	:	specific blanking
adi_t_gb	:	general blanking
boundary_interpolation	:	interface boundary interpolation
bcs_*	:	boundary conditions for physical boundaries
des_*	:	turbulence equations for SA model and DES
residual	:	calculate residual between pseudo-time steps
init_*	:	specify initial guess for the solver

To run the solver, many options, such as artificial dissipation schemes, amount of artificial dissipation, pseudo-time step, physical time-step, ADI or Runge-Kutta, etc., have to be specified. This can be done by changing the input file "control.dat". The following is the format of control.dat:

```
read(1,*) nzone,ng,i01,i02
read(1,*) istart,itmax,it_int,ns,nstop,icycle,dimf
read(1,*) irk,(alfa(i),i=1,irk)
read(1,*) nsch1,nsch2,idcv,ikw,ifso,iardis
do nz=1,nzone
read(1,*) img(1,nz),jmg(1,nz),kmg(1,nz)
end do
do nz=1,nzone
read(1,*) icrs_g(nz),jcrs_g(nz),kcrs_g(nz)
end do
do n=1,ng
read(1,*) iter(n),itm(n)
end do
do nz=1,nzone
read(1,*) (eps_global(i,nz),i=1,6)
end do

do nz=1,nzone
read(1,*) (ep_rk_global(i,nz),i=1,3)
end do
do nz=1,nzone
read(1,*) (ep_pr_global(i,nz),i=1,3)
end do
read(1,*) dtime,mtime,it_out
read(1,*) latu,ren,cfl1,vnn1,cfl2,vnn2,beta
```

```
pdiss_coef=eps_global(1,1)*beta
read(1,*) ckin,cein,cwin,coefv,alfa1
do nz=1,nzone
read(1,*) (kgs_g(i,nz),i=1,6)
end do
read(1,*) ip,jp,kp,np
do nz=1,nzone
read(1,*) nzblanking(nz)
if(nzblanking(nz).ne.0)then
read(1,*) ((blanking(i,j,1,nz),j=1,2),i=1,3)
end if
end do
```

Variables:

****

Line 1:

```
read(1,*) nzone,ng,i01,i02
```

nzone: number of blocks

ng: multi-grid levels (just set it to 1 since mostly we only need single-grid calculation)

i01, i02: any value(useless)

****

Line 2:

```
read(1,*) istart,itmax,it_int,ns,nstop,icycle,dimf
```

istart: ==0 start from scratch

==1 start from previous calculation

```
itmax:    maximum pseudo-time iteration steps for a real time step advancing
usually a number around 15
it_int:   any value
ns:       designed for multi-grid calculation, use 1
nstop:    same as ns
icycle:   same as ns
dimf:     any value

**********************************************************************
Line 3:
read(1,*) irk,(alfa(i),i=1,irk)

irk:      use 1 for ADI scheme, use 4 for Runga-Kutta scheme
alfa:     use 1. for ADI scheme, use

**********************************************************************
Line 4:
read(1,*) nsch1,nsch2,idcv,ikw,ifso,iardis

nsch1:    use 1
nsch2:    any value
idcv:     use 1 for central difference scheme
ikw:     any value
ifso:     any value
iardis:   use 1 for matrix dissipation
use 3 for QUICK scheme

**********************************************************************
```

```
Line 5:
do nz=1,nzone
read(1,*) img(1,nz),jmg(1,nz),kmg(1,nz)
end do

imax, jmax, kmax for blocks from 1 to nzone

**********************************************************************
Line 6:
do nz=1,nzone
read(1,*) icrs_g(nz),jcrs_g(nz),kcrs_g(nz)
end do

all variables here are designed for multi-grid calculation, use 1
for all of them

**********************************************************************
Line 7:
do n=1,ng
read(1,*) iter(n),itm(n)
end do



**********************************************************************
Line 8:
do nz=1,nzone
read(1,*) (eps_global(i,nz),i=1,6)
end do
```

coefficients for artificial dissipation, use 0.01 or 0.004 for all of them

****

Line 9:

do nz=1,nzone

read(1,*) (ep_rk_global(i,nz),i=1,3)

end do

designed for Runga-Kutta integration, use 0.5 for all of them. Actually for ADI scheme, the values of these arrays don't matter.

****

Line 10:

do nz=1,nzone

read(1,*) (ep_pr_global(i,nz),i=1,3)

end do

Same as line 9.

****

Line 11:

read(1,*) dtime,mtime,it_out

dtime: Real time step (non-dimensionalized), this value should depend on calculation. If dtime == very large number, such as 1.e24, it means the calculation is running in a steady solution

manner. While dtime lies between [0:0.5], the solver is running in a time-accurate fashion.

mtime: specify total number of real time steps

it_out: specify the frequency for time-accurate solution recording, e.g. if it_out == 5, then the solution will be written for every 5 time steps.

****

Line 12:

read(1,*) latu,ren,cfl1,vnn1,cfl2,vnn2,beta

latu: ==1, turbulent flow

==0, Laminar flow

ren: Reynolds nubmer of the calculation

cfl1: CFL number for the NS equations, for ADI scheme, some value between 3 and 5 should work fine

vnn1: von Neumann number, 1.5~2.5

cfl2, vnn2: for turbulent flow calculation

beta: coefficient for artificial compressibility method, use 1.

****

Line 13

```
read(1,*) ckin,cein,cwin,coefv,alfa1

ckin, cein, cwin, coefv, alfa1: all designed for turbulent flow
calculation

**********************************************************************
Line 14:
do nz=1,nzone
read(1,*) (kgs_g(i,nz),i=1,6)
end do

kgs_g: specify boundary conditions

the first dimension of kgs_g(i,nz) corresponding to the six
boundaries of a specific block, which mean the imin, imax,
jmin, jmax, kmin, kmax boundary respectively.

== 1    solid wall
== 2  periodic condition
== 3  Free surface
== 4  exit
== 5  inlet

**********************************************************************
Line 15:
read(1,*) ip,jp,kp,np

specify pressure fix point, (ip, jp, kp, np) means the ip, jp, kp
point of block np. Please note to pick a point on the plane next to
```

```
the exit plane.


**********************************************************************
Line 16:
do nz=1,nzone
read(1,*) nzblanking(nz)
if(nzblanking(nz).ne.0)then
read(1,*) ((blanking(i,j,1,nz),j=1,2),i=1,3)
end if
end do


designed for specific blanking


If no blanking is required, uese 0 for all nzblanking.
```

The output of the solver is stored in files named as "solu?????" where the last five digits are the physical time step number. This can be transfered into ".plt" format with the help of post processor.

B.4 Post-processor

The post-processor utilize Tecplot library to directly transfer the output file into ".plt" file. It reads in grid.dat and the solution file and then combine these files together into a single ".plt" file. The post-processor is named as "mktec.f90". It can be compiled with the following line:

ifort -o mktec mktec.f90 $TEC100HOME/lib/tecio.a -Vaxlib

Since the post-processor requires the tecplot library file, thus it's only supported on the Linux machines that have tecplot installed. To run the post-process, use "mktec solu?????" where "solu?????" is the output file from the kernel solver. The output file is named as "solu?????.plt". Remember that in "mktec.f90", the number of zones (nzone variable) needs to be changed according to the actual number of zones used for the calculation.

REFERENCES

Abgrall, R. (2001), "Toward the ultimate conservative scheme: Following the quest," *Journal of Computational Physics*, 167(2), 277–315.

Abgrall, R. and Mezine, M. (2003), "Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems," *Journal of Computational Physics*, 188(1), 16–55.

Ali, K. H. M. and Karim, O. (2002), "Simulation of flow around piers," *Journal of Hydraulic Research*, 40(2), 161–174.

Aoki, T. (2001), "3D simulation for falling papers," *Computer Physics Communications*, 142(1-3), 326–329.

Athanasiadis, A. and Deconinck, H. (2003), "Object-oriented three-dimensional hybrid grid generation," *International Journal for Numerical Methods in Engineering*, 58(2), 301–318.

Babushka, I. and Aziz, A. K. (1976), "On the angle condition in the finite-element method," *SIAM Journal of Numerical Analysis*, 13(6).

Baggett, J. S., Jimenez, J., and Kravchenko, A. G. (1997), "Resolution requirements in large-eddy simulations of shear flows," *Center for Turbulence Research Annual Research Briefs*, pages 51–66.

Balaras, E. and Benocci, C. (1994), "Subgrid-scale models in finite-difference simulations of complex wall bounded flows," *AGARD CP 551*, 2.1.

Balaras, E., Benocci, C., and Piomelli, U. (1996), "Two-layer approximate boundary conditions for large-eddy simulation," *AIAA Journal*, 34, 1111.

Baldwin, J., Tarbell, J., and Deutsch, S. (1991), "Mean velocities and Reynolds stresses within regurgitant jets produced by tilting valves," *American Society of Artificial Internal Organs Transactions*, 37.

Beam, R. M. and Warming, R. F. (1976), "An implicit finite-difference algorithm for hyperbolic systems in conservation-law form," *Journal of Computational Physics*, 22, 87–110.

Bell, J., Colella, P., and Trangenstein, J. (1989), "Higher-order Godunov methods for general systems of hyperbolic conservation-laws," *Journal of Computational Physics*, 82(2), 362–397.

Benek, J. A., Steger, J. L., and Dougherly, F. C. (1983), "A flexible grid embedding technique with application to the Euler equations," *AIAA paper 83-1944*.

Black, M. and Drury, P. (1994). "Mechanical and other problems of artificial valves," . In Berlin, B. C., editor, *Current Topics in Pathology*, pages 128–159. Springer-Verlag.

Blanco, M. and Zingg, D. (1998), "Fast Newton-Krylov method for unstructured grids," *AIAA Journal*, 36(4), 607–612.

Bluestein, D., Li, Y., and Krukenkamp, I. (2002), "Free emboli formation in the wake of bi-leaflet mechanical heart valves and the effects of implantation techniques," *Journal of Biomechanics*, 35(12), 1533–1540.

Bosch, G. and Rodi, W. (1998), "Simulation of vortex shedding past a square cylinder with different turbulence models," *International Journal of Numerical Methods in Fluids*, 28(4), 601–616.

Brown, C., Leverett, L., and Lewis, C. (1975), "Morphological, biological, and functional changes in human platlets subjet to shear stress," *Journal of Laboratory and Clinical Medicine*, 86(3).

Brown, P. and Saad, Y. (1990), "Hybrid Krylov methods for nonlinear-systems of equtions," *SIAM Journal on Scientific and Statistical Computing*, 11(3), 450–481.

Burton, T. and Eaton, J. (2002), "Analysis of a fractional-step method on overset grids," *Journal of Computational Physics*, 177(2), 336–364.

Caughey, D. A. and Jameson, A. (2003), "Fast preconditioned multigrid solution of the Euler and Navier-Stokes equations for steady, compressible flows," *International Journal of Heat and Fluid Flows*, 43(5), 537–553.

Cebeci, T. and Bradshaw, P. (1977). *Momentum Transfer in Boundary Layers*. Hemisphere.

Chandran, K. B., Cabell, G. N., and Khalighi, B. (1983), "Laser anemometry measurements of pulsatile flow past aortic valve prosthesis," *Journal of Biomechanics*, 16(10), 865–873.

Chandran, K. B., Dexter, E. U., and Aluri, S. (1998), "Negative pressure transients with mechanical heart-valve closure: correlation between in vitro and in vivo results," *Annals of Biomedical Engineering*, 26, 546–556.

Chandran, K. B., Khalighi, B., and Chen, C. J. (1985), "Experimental study of physiological pulsatile flow past valve prosthesis in a model of human aorta–II. Tilting disk valves and the effect of orientation," *Journal of Biomechanics*, 18(10), 773–780.

Chandran, K. B., Lee, C. S., and Chen, L. D. (1994), "Pressure field in the vicinity of mechanical valve occluders at the instant of valve closure: correlation with cavitation initiation," *Journal of Heart Valve Disease*, 3 Suppl 1, S65–75.

Chapman, D. R. (1979), "Computational aerodynamics development and out-look," *AIAA Journal*, 17, 1293.

Chen, H. and Liu, T. (1999), "Turbulent flow induced by full-scale ship in harbor," *Journal of Engineering Mechanics-ASCE*, 125(7), 827–835.

Chen, H.-C. and Liu, T. (2000), "Chimera RANS simulation of ship and fender coupling for berthing operations," *International Journal of Offshore and Polar Engineering*, 10(2), 112–122.

Chesshire, G. and Henshaw, M. D. (1990), "Composite overlapping meshes for the solution of partial differential equations," *Journal Computational Physics*, 90, 1–64.

Choi, S. (1999), "Note on the use of momentum interpolation method for unsteady flows," *Numerical Heat Transfer Part A-Applications*, 36(5), 545–550.

Choi, S. and Kang, H. (2004), "Reynolds stress modeling of vegetated open-channel flows," *Journal of Hydraulic Research*, 42(1), 3–11.

Chorin, A. (1967), "Numerical solution of Navier-Stokes equations for an imcompressible fluid," *Bulletin of the American Mathematical Society*, 73(6), 928–&.

Chorin, A. (1968), "Numerical solution of Navier-Stokes equations," *Mathematics of Computation*, 22(104), 745.

Chrisohoides, A., Sotiropoulos, F., and Sturm, T. W. (2003), "Coherent structures in flat–bed abutment flow: Computational fluid dynamics simulations and experiments," *Journal of Hydraulic Engineering*, 129(3), 177–186.

Constantinescu, G. and Squires, K. (2003), "LES and DES investigations of turbulent flow over a sphere at Re=10,000," *Flow Turbulence and Combustion*, 70(1-4), 267–298.

Constantinescu, G. and Squires, K. (2004), "Numerical investigations of flow over a sphere in the subcritical and supercritical regimes," *Physics of Fluids*, 16(5), 1449–1466.

Dargahi, B. (1989), "The turbulent flow field around a circular cylinder," *Experiments in Fluids*, 8, 1–12.

Dargahi, B. (1990), "Controlling mechanism of local scouring," *Journal of Hydraulic Engineering*, 116(10), 1197–1214.

Darwish, M. and Moukalled, F. (2003), "TVD schemes for unstructured grids," *International Journal of Heat and Mass Transfer*, 46(4), 599–611.

Davidson, L. (1996), "A pressure correction method for unstructured meshes with arbitrary control volumes," *International Journal for Numerical Methods In Fluids*, 22(4), 265–281.

Djomehri, M. and Biswas, R. (2003), "Performance enhancement strategies for multi-block overset grid cfd applications," *Parallel Computing*, 29(11-12), 1791–1810.

Dong, S. and Karniadakis, G. (2004), "Dual-level parallelism for high-order CFD methods," *Parallel Computing*, 30(1), 1–20.

Douglass, R., Carey, G., White, D., Hansen, G., Kallinderis, Y., and Weatherill, N. (2002), "Current views on grid generation: Summaries of a panel discussion," *Numerical Heat Transfer Part B-Fundamentals*, 41(3-4), 211–237.

Durbin, P. A. (1995), "Seperated flow computations using $k-\varepsilon-v^2$," *AIAA Journal*, 33(4), 659.

Ekici, K. and Lyrintzis, A. (2003), "A parallel Newton-Krylov method for Navier-Stokes rotorcraft codes," *International Journal of Computational Fluid Dynamics*, 17(3), 225–230.

Ellis, J., Travis, B., and Yoganathan, A. (2000), "An in vitro study of the hinge and near-field forward flow dynamics of the St. jude Medical (R) Regent (TM) bileaflet mechanical heart valve," *Annals of Biomedical Engineering*, 28(5), 524–532.

Ellis, J. and Yoganathan, A. (2000), "A comparison of the hinge and near-hinge flow fields of the St Jude Medical Hemodynamic Plus and Regent bileaflet mechanical heart valves," *Journal of Thoracic and Cardiovascular Surgery*, 119(1), 83–93.

Ellis, J. T., *An in vitro investigation of the leakage and hinge flow fields through bileaflet mechanical heart valves and their relevance to thrombogenesis*. PhD thesis, Georgia Institute of Technology (1999).

Ellis, J. T., Healy, T. M., and Fontaine, A. A. (1996a), "An in vitro investigation of the retrograde flow fields of two bileaflet mechanical heart valves," *Journal of Heart Valve Disease*, 5(6), 600–606.

Ellis, J. T., Healy, T. M., and Fontaine, A. A. (1996b), "Velocity measurements and flow patterns within the hinge region of Medtronic Parallel bileaflet mechanical valve with clear housing," *Journal of Heart Valve Disease*, 5(6), 591–599.

Ellis, J. T. and Yogananthan, A. P. (2000), "A comparison of the hinge and near-hinge flow fields of the St. Jude medical hemodynamics plus and regent bileaflet mechanical heart valves," *Journal of Thorac Cardiovasc Surgery*, 119(1), 83–93.

Fatemi, R. and Chandran, K. B. (1989), "An in vitro comparative study of St. Jude Medical and Edwards-Duromedics bileaflet valves using laser anemometry," *Journal of Biomechanical Engineering*, 111(4), 298–302.

Flynn, M. and Eisner, A. (2004), "Verification and validation studies of the time-averaged velocity field in the very near-wake of a finite elliptical cylinder," *Fluid Dynamics Research*, 34(4), 273–288.

Fontaine, A., He, S., and Stadter, R. (1996), "In vitro assessment of prosthetic valve funciton in mitral valve replacement with chordal preservation techniques," *Journal of Heart Valve Disease*, 5(2), 186–198.

Forsythe, J., Squires, K., Wurtzler, K., and Spalart, P. (2004), "Detached-eddy simulation of the F-15E at high alpha," *Journal of Aircraft*, 41(2), 193–200.

Fung, Y. C. (1984). *Biodynamics: circulation*. New York: Springer-Verlag.

Ge, L., Jones, S., Sotiropoulos, F., Healy, T., and Yoganathan, A. (2003), "Numerical simulation of flow in mechanical heart valves: Grid resolution and the assumption of flow symmetry," *Journal of Biomechanical Engineering-Transactions of The ASME*, 125(5), 709–718.

Ge, L., Lee, S. O., Sotiropoulos, F., and Sturm, T. (2004), "3D unsteady RANS modeling of complex hydraulic engineering flows. Part II: Model calibration and flow physics," *Journal of Hydraulic Engineering*. submitted.

Ge, L. and Sotiropoulos, F. (2004), "3D unsteady RANS modeling of complex hydraulic engineering flows. Part I: Numerical model," *Journal of Hydraulic Engineering*. submitted.

Germano, M., Piomelli, U., and Moin, P. (1991), "A dynamic subgrid-scale eddy viscosity model," *Physics of Fluids A-Fluid Dynamics*, 3(7), 1760–1765.

Gilmanov, A., Sotiropoulos, F., and Balaras, E. (2003), "A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids," *Journal of Computational Physics*, 191(2), 660–669.

Graf, T., Reul, H., and Dietz, W. (1992), "Cavitation of mechanical heart valves under physiologic conditions," *Journal of Heart Valve Disease*, 1(1), 131–141.

Gropp, W., Kaushik, D., Keyes, D., and Smith, B. (2001), "High-performance parallel implicit CFD," *Parallel Computing*, 27(4), 337–362.

Harlow, F. H. and Welch, J. E. (1965), "Numerical calculation of time-dependent viscous incompressible flows with free surface," *Physics of Fluids*, 8, 2182–2189.

Harten, A., Lax, P. D., and Van Leer, B. (1983), "On upstream differencing and Gudonov-type schemes for hyperbolic conservation laws," *SIAM Reviews*, 25(1), 35–61.

Hedges, L. S., Travin, A. K., and Spalart, P. R. (2002), "Detached-Eddy Simulations over a simplified landing gear," *Journal of Fluids Engineering -Transactions of the ASME*, 124(2), 413–423.

Hinatsu, M. and Ferziger, J. (1991), "Numerical computation of unsteady incompressible-flow in complex-geometry using a composite multigrid technique," *International Journal for Numerical Methods in Fluids*, 13(8), 971–997.

Hirsch, C. (1999). "Industrial computational fluid dynamics," . In *Industrial Computational Fluid Dynamics, Lecture Series 1999-06*. von Karman Institute for Fluid Dynamics.

Hollanders, H., Lerat, A., and Peyret, R. (1985), "Three-dimensional calculations of transonic viscous flows by and implicit method," *AIAA Journal*, 23, 1670.

Huang, Z. J., Merkle, C. L., Abdallah, S., and Tarbell, J. M. (1994), "Numerical simulation of unsteady laminar flow through a tilting disk heart valve: prediction of vortex shedding," *Journal of Biomechanics*, 27(4), 391–402.

Hubbard, B. and Chen, C. H. (1990). *A Chimera scheme for incompressible viscous flows with application to submarine hydrodynamics*. Technical Report 94-2210, AIAA paper.

Hussain, A. K. F. M. and Reynolds, W. C. (1972), "The mechanics of an organized wave in turbulent shear flow. Part 2: experimental results," *Journal of Fluid Mechanics*, 54, 241–261.

Iaccarino, G. and Durbin, P. A. (2000). *Unsteady 3D RANS Simulations Using $v^2 - f$ Model*. Technical report, Center for Turbulence Research.

Iaccarino, G., Ooi, A., Durbin, P., and Behnia, M. (2003), "Reynolds averaged simulation of unsteady separated flow," *International Journal of Heat and Fluid Flow*, 24(2), 147–156.

Jameson, A. and Turkel, E. (1981), "Implicit schemes and LU-decompositions," *Mathematics of Computation*, 37(156), 385–397.

Jin, H., Frumkin, M., and Yan, J. (2000), "Automatic generation of OpenMP directives and its application to computational fluid dynamics codes," *High Performance Computing, Proceedings*, 1940, 440–456.

Jones, R., Harvey, A., and Acharya, S. (2001), "Two-equation turbulence modeling for impeller stirred tanks," *Journal of Fluids Engineering-Transactions of The ASME*, 123(3), 640–648.

Jones, W. P. and Launder, B. E. (1972), "Prediction of laminarization with a 2-equation model of turbulence," *International Journal of Heat and Mass Transfer*, 15(2), 301.

Kafesjian, R., Howanec, M., and Ward, G. (1994), "Cavitation damage of Pyrolytic Carbon in mechanical heart valves," *Journal of Heart Valve Disease*, 3(Supp 1), S2–S7.

Khawaja, A. and Kallinderis, Y. (2000), "Hybrid grid generation for turbomachinery and aerospace applications," *International Journal for Numerical Methods in Engineering*, 49(1-2), 145–166.

Khier, W., Breuer, M., and Durst, F. (2000), "Flow structure around trains under side wind conditions: a numerical study," *Computers & Fluids*, 29(2), 179–195.

Kim, D. and Choi, H. (2000), "A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids," *Journal of Computational Physics*, 162(2), 411–428.

Kim, J. and Moin, P. (1985), "Application of a fractional-step method to incompressible Navier-Stokes equations," *Journal of Computational Physics*, 59(2), 308–323.

Kimura, I. and Hosoda, T. (2003), "A non-linear k-ε model with realizability for prediction of flows around bluff bodies," *International Journal for Numerical Methods in Fluids*, 42(8), 813–837.

King, M. J., Corden, J., David, T., and Fisher, J. (1996), "A three-dimensional, time-dependent analysis of flow through a bileaflet mechanical heart valve: Comparison of experimental and numerical results," *Journal of Biomechanics*, 29(5), 609–618.

King, M. J., David, T., and Fisher, J. (1994), "An initial parametric study on fluid flow through bileaflet mechanical heart valves using computational fluid dynamics," *Journal of Engineering in Medicine*, 208, 63–72.

King, M. J., David, T., and Fisher, J. (1997), "Three-dimensional study of the effect of two leaflet opening angles on the time-dependent flow through a bileaflet mechanical heart valve," *Medical Engineering and Physics*, 19(5), 235–241.

Kirkpatrick, M. P., Armfield, S. W., and Kent, J. H. (2003), "A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid," *Journal of Computational Physics*, 184, 1–36.

Kish, L. B. (2002), "End of Moore's law: thermal (noise) death of integration in micro and nano electronics," *Physics Letters A*, 305(3-4), 144–149.

Klepetko, W. and Moritz, A. (1989), "Leaflet fracture in Edwards-Duromedic bileaflet valves," *Journal of Thoracic and Cardiovascular Surgery*, 97, 90–94.

Knoll, D. and Keyes, D. (2004), "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, 193(2), 357–397.

Kobayashi, M., Pereira, J., and Pereira, J. (1999), "A conservative finite-volume second-order-accurate projection method on hybrid unstructured grids," *Journal of Computational Physics*, 150(1), 40–75.

Koumoutsos, A., Tourlidakis, A., and Elder, R. (2000), "Computational studies of unsteady flows in a centrifugal compressor stage," *Proceedings of the Institution of Mechanical Engineers Part A-Journal of Power And Energy*, 214(A6), 611–633.

Kremenetsky, M., Tysinger, T., and Posey, S. (2000). "Considerations for parallel CFD enhancements on SGI ccNUMA and cluster architectures," . In *Proc. of the Parallel CFD 2000 Conference*, Trondheim, Norway.

Lambropoulos, N., Koubogiannis, D., and Giannakoglou, K. (2004), "Acceleration of a Navier-Stokes equation solver for unstructured grids using agglomeration multigrid and parallel processing," *Computer Methods in Applied Mechanics And Engineering*, 193(9-11), 781–803.

Lamson, T., Rosenberg, G., and Geselowitz, D. (1993), "Relative blood damage in the three phases of a prosthetic heart valve flow cycle," *ASAIO Journal*, 39, 626–633.

Lee, C. S., Chandran, K. B., and Chen, L. D. (1994), "Cavitation dynamics of mechanical heart valve prostheses," *Artificial Organs*, 18, 758–767.

Leo, H., He, Z., Ellis, J., and Yoganathan, A. (2002), "Microflow fields in the hinge region of the carbomedics bileaflet mechanical heart valve design," *Journal of Thoracic and Cardiovascular Surgery*, 124(3), 561–574.

Leschziner, M. A. (2000). "Statistical turbulence modelling for the computation of physically complex flows," . In Lesieur, M., Yaglom, A., and David, F., editors, *New trends in turbulence*. Springer.

Lilly, D. K. (1965), "On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems," *Monthly Weather Review*, 93(1), 11.

Lin, F. B. and Sotiropoulos, F. (1997a), "Assessment of artificial dissipation models for three-dimensional incompressible flow solutions," *Journal of Hydraulic Engineering*, 119(2), 331–340.

Lin, F. B. and Sotiropoulos, F. (1997b), "Strongly coupled multigrid method for 3-D incompressible flows using near-wall turbulence closures," *Journal of Hydraulic Engineering*, 119(2), 314–324.

Lowrie, R. (2004), "A comparison of implicit time integration methods for nonlinear relaxation and diffusion," *Journal of Computational Physics*, 196(2), 566–590.

Mahesh, K., Constantinescu, G., and Moin, P. (2004), "A numerical method for large-eddy simulation in complex geometries," *Journal of Computational Physics*, 197(1), 215–240.

Marcum, D. (2001), "Efficient generation of high-quality unstructured surface and volume grids," *Engineering With Computers*, 17(3), 211–233.

Marcum, D. L. and Weatherill, N. P. (1995), "Unstructured grid generation using iterative point insertion and local recoonection," *AIAA Journal*, 33(9), 1619–1625.

Martin, M., Parada, M., and Doallo, R. (2004), "High performance air pollution simulation using OpenMP," *Journal of Supercomputing*, 28(3), 311–321.

Meakin, R. L. (1995), "The chimera method of simulation for unsteady three-dimensional viscous flow," *Computational Fluid Dynamics Review*, pages 70–86.

Melville, B. W. (1997), "Pier and abutment scour: Integrated approach," *Journal of Hydraulic Engineering*, 123(2), 125–136.

Merkle, C. L. and Athavale, M. (1987), "A time-accurate unsteady incompressible flow algorithm based on artificial compressibility," *AIAA Paper 87-1137*.

Meyer, R. S., Deutsch, S., and Maymir, J. C. (1997), "Three-component laser Doppler velocimetry measurements in the regurgitant flow region of a Bjork-Shiley monostrut mitral valve," *Annals of Biemedical Engineering*, 25(6), 1081–1091.

Mittal, R. and Moin, P. (1997), "Suitability of upwind-biased finite difference schemes for Large-Eddy simulation of turbulent flows," *AIAA Journal*, 35(8), 1415–1417.

Mittal, R., Simmons, S., and Najjar, F. (2003), "Numerical study of pulsatile flow in a constricted channel," *Journal of Fluid Mechanics*, 485, 337–378.

Mittal, R., Simmons, S., and Udaykumar, H. (2001), "Application of large-eddy simulation to the study of pulsatile flow in a modeled arterial stenosis," *Journal of Biomechanical Engineering-Transactions of the ASME*, 123(4), 325–332.

Mohands, N., Hochmuth, R., and Spaeth, E. (1974), "Adhesions of red cells to foreign surfaces in the presence of flow," *Journal of Biomedical Materials Research*, 8, 119–136.

Moin, P. and Mahesh, K. (1998), "Direct numerical simulation: A tool in turbulence research," *Annual Review of Fluid Mechanics*, 30, 539–578.

Moinier, P., Muller, J., and Giles, M. (2002), "Edge-based multigrid and preconditioning for hybrid grids," *AIAA Journal*, 40(10), 1954–1960.

Moore, G. E. (1965), "Cramming more components onto integrated circuits," *Electronics*, 38(8).

Nakayama, A. and Miyashita, K. (2001), "URANS simulation of flow over smooth topography," *International Journal of Numerical Methods for Heat and Fluid Flow*, 11(8), 723–743.

Nemec, M. and Zingg, D. (2002), "Newton-Krylov algorithm for aerodynamic design using the Navier-Stokes equations," *AIAA Journal*, 40(6), 1146–1154.

Ollivier-Gooch, C. (2003), "Coarsening unstructured meshes by edge contraction," *International Journal for Numerical Methods in Engineering*, 57(3), 391–414.

Olsen, N. R. B. (2003), "3D CFD modeling of a self–forming meandering channel," *Journal of Hydraulic Engineering*, 129(5), 366–372.

Olsen, N. R. B. and Kjellesvig, H. M. (1998), "Three-dimensional numerical flow modeling for estimation of maximum local scour depth," *Journal of Hydraulic Research*, 36(4), 579–590.

Owen, S. and Saigal, S. (2000), "H-morph: an indirect approach to advancing front hex meshing," *International Journal for Numerical Methods in Engineering*, 49(1-2), 289–312.

Paik, J. C., Ge, L., and Sotiropoulos, F. (2004), "Toward the simulation of complex 3D shear flows using unsteady statistical turbulence models," *International Journal of Heat and Fluid Flow*, 25(3), 513–527.

Pan, H. and Damodaran, M. (2002), "Parallel computation of viscous incompressible flows using Godunov-projection method on overlapping grids," *International Journal for Numerical Methods in Fluids*, 39(5), 441–463.

Papanicolaou, A. N., Diplas, P., Evaggelopoulos, N., and Fotopoulos, S. (2002), "Stochastic incipient motion criterion for spheres under various bed packing conditions," *Journal of Hydraulic Engineering*, 128, 369–380.

Peskin, C. S. (1972), "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, 10, 252–271.

Piomelli, U. (1999), "Large-eddy simulation: achievements and challenges," *Progress in Aerospace Sciences*, 35(4), 335–362.

Piomelli, U. and Balaras, E. (2002), "Wall–layer models for Large–Eddy Simulations," *Annual Review of Fluid Mechanics*, 34, 349–374.

Piomelli, U., Balaras, E., Pasinato, H., Squires, K. D., and Spalart, P. R. (2003), "The inner-outer layer interface in large-eddy simulations with wall-layer models," *International Journal of Heat and Fluid Flow*, 24, 538–550.

Pope, S. B. (2004), "Ten questions concerning the large-eddy simulation of turbulent flows," *New Journal of Physics*, 6(35).

Rai, M. M. (1986), "A conservative treatment of zonal boundary scheme for Euler equations calculations," *Journal of Computational Physics*, 62, 472–503.

Richmond-Bryant, J. (2003), "Verification testing in computational fluid dynamics: an example using Reynolds-averaged Navier-Stokes methods for two-dimensional flow in the near wake of a circular cylinder," *International Journal for Numerical Methods in Fluids*, 43(12), 1371–1389.

Rodi, W. (1993), "Turbulence models and their applications in hydraulics," *IAHR monograph.*

Roe, P. L. (1981), "Approximate Riemann sovlers, parameter vectors, and difference schemes," *Journal of Computational Physics*, 43, 357.

Rogers, S., Kwak, D., and Kiris, C. (1991), "Steady and unsteady solutions of the incompressible Navier-Stokes equations," *AIAA Journal*, 29(4), 603–610.

Rogers, S., Roth, K., Cao, H., Nash, S., and Baker, M. (2001), "Computation of viscous flow for a Boeing 777 aircraft in landing configuration," *Journal of Aircraft*, 38(6), 1060–1068.

Rosenfeld, M., Kwak, D., and Vinokur, M. (1991), "A fractional step solution method for the unsteady incompressible Navier-Stokes equations in generalized coordinate systems," *Journal of Computational Physics*, 94(1), 102–137.

Sallam, A. and Hwang, N. (1984), "Human red blood cell hemolysis in a turbulent shear flow: contribution of Reynolds shear stresses," *Biorheology*, 21, 783–797.

Schoephoerster, R. and Chandran, K. (1991), "Velocity and turbulence measurements past mitral-valve prostheses in a model left-ventricle," *Journal of Biomechanics*, 24(7), 549–562.

Schwartz, H. A. (1869), "Uber einige abbildungsaufgaben," *Ges. Abh.*

Shim, E.-B. and Chang, K.-S. (1997), "Numerical analysis of three-dimensional Björk–Shiley valvular flow in an aorta," *Journal of Biomechanical Engineering*, 119, 45–51.

Shur, M., Spalart, P. R., Strelets, M., and Travin, A. (1999). "Detached-Eddy Simulation of an airfoil at high angle of attach," . In Rodi, W. and Laurence, D., editors, *Proceedings of 4th International Symposium of Engineering Turbulence Modelling and Measurements*, pages 669–678. Elsevier.

Sinha, S. K., Sotiropoulos, F., and Odgaard, A. J. (1998), "Three-dimensional numerical model for flow through natural rivers," *Journal of Hydraulic Engineering*, 124(1), 13–24.

Smith, R., Blick, E., Coalson, J., and Stein, P. (1972), "Thrombus production by turbulence," *Journal of Applied Physiology*, 32, 261–264.

Sorensen, K., Hassan, O., Morgan, K., and Weatherill, N. (2003), "A multigrid accelerated time-accurate inviscid compressible fluid flow solution algorithm employing mesh movement and local remeshing," *International Journal for Numerical Methods in Fluids*, 43(5), 517–536.

Sotiropoulos, F. (2000). "Progress in modeling 3-D shear flows using RANS equations and advanced turbulence closures," . In *Calculation of Complex Turbulent Flows, Advances in Fluid Mechanics Series*, chapter 6, pages 209–248. WIT Press.

Sotiropoulos, F. and Abdallah, S. (1991), "The discrete continuity equation in primitive variable solutions of incompressible-flow," *Journal of Computational Physics*, 95(1), 212–227.

Spalart, P. R. (2000), "Strategies for turbulence modelling and simulations," *International Journal of Heat and Fluid Flow*, 21(3), 252–263.

Spalart, P. R. and Allmaras, S. R. (1994), "A one-equation turbulence model for aerodynamics flows," *La Recherche Aerospatiale*, 1, 5–21.

Spalart, P. R., Jou, W. H., Strelets, M., and Allmaras, S. R. (1997). "Comments on the feasibility of LES for wings, and on hybrid RANS/LES approach," . In Liu, C. and Liu, Z., editors, *First AFOSR Inetrnational Conference on DNS/LES*, Ruston, LA.

Speziale, C., Sarkar, S., and Gatski, T. (1991), "Modeling the pressure strain correlation of turbulence - an invariant dynamic-systems approach," *Journal of Fluid Mechanics*, 227, 245–272.

Squires, K., Forsythe, J., and Spalart, P. (2001). "Detached-eddy simulation of the seperated flor around a forebody cross section," . In Geurts, B., Friedrich, R., and Metais, O., editors, *Direct and Large-Eddy Simulation IV*, volume 8 of *ERCOFTAC Series*, pages 481–500. Kluwer Academic Press.

Steger, J. L. and Benek, J. A. (1987), "On the use of composite grid schemes in computational aerodynamics," *Computer Methods in Applied Mechanics and Engineering*, 64, 301–320.

Steger, J. L. and Warming, R. F. (1979). *Flux vector splitting of the inviscid gasdynamics equations with applications to finite difference methods*. Technical report, NASA TM 78605.

Stein, P. D. and Sabbah, M. N. (1974), "Measured turbulence and its effect on thrombus formation," *Circulation Research*, 35, 608–614.

Stevenson, D. and Yoganathan, A. (1985), "Numerical-simulation of steady turbulent-flow through trileaflet aortic heart-valves .1. computational scheme and methodology," *Journal of Biomechanics*, 18(12), 899–907.

Stevenson, D., Yoganathan, A., and Williams, F. (1985), "Numerical-simulation of steady turbulent-flow through trileaflet aortic heart-valves .2. results on 5 models," *Journal of Biomechanics*, 18(12), 909.

Sturm, T., Sotiropoulos, F., Landers, M., Gotvald, T., Lee, S., Ge, L., Navarro, R., and Escauriaza, C. (2004). *Laboratory and 3D numerical modeling with field monitoring of regional bridge scour in georgia*. Technical Report GDOT Research Project No. 2002, Georgia Department of Transportation.

Subramanian, A., Mu, H., Kadambi, J., Wernet, M., Brendzel, A., and Harasaki, H. (2000), "Particle image velocimetry investigation of intra-valvular flow fields of a bileaflet mechanical heart valve in a pulsatile flow," *Journal of Heart Valve Disease*, 9(5), 721–731.

Sutera, S., Croce, P., and Mehrjardi, M. (1972), "Hemolysis and subhemolytic alterations of human RBC induced by turbulence shear stress," *Transanctions of the American Society of Artificial Internal Organs*, 18, 335–341.

Takahashi, D., Sato, M., and Boku, T. (2003), "An OpenMP implementation of parallel FFT and its performance on IA-64 processors," *Openmp Shared Memory Parallel Programming*, 2716, 99–108.

Takaki, R., Yamamoto, K., Yamane, T., Enomoto, S., and Mukai, J. (2003), "The development of the UPACS CFD environment," *High Performance Computing*, 2858, 307–319.

Takanashi, S. and Takemoto, M. (1995), "An automatic grid generation procedure for complex aircraft configurations," *Computers & Fluids*, 24(4), 393–400.

Tang, H. S., *Domain Decomposition Algorithms for 3D Unsteady Incompressible Flows*. PhD thesis, Georgia Institute of Technology (2001).

Tang, H. S., Jones, S. C., and Sotiropoulos, F. (2003), "An overset–grid method for 3D unsteady incompressible flows," *Journal of Computational Physics*, 191(2), 567–600.

Temam, R. (1969), "Approximation to a solution of Navier-Stokes equations by a fractional step method .2." *Archive for Rational Mechanics and Analysis*, 33(5), 377.

Travin, A., Shur, M., Strelets, M., and Spalart, P. (2000), "Detached-eddy simulations past a circular cylinder," *Flow Turbulence and Combustion*, 63(1-4), 293–313.

Tseng, M., Yen, C., and Song, C. C. S. (2000), "Computation of three-dimensional flow around square and circular piers," *International Journal for Numerical Methods in Fluids*, 34, 207–227.

Tucker, P. G. and Pan, Z. (2001), "URANS computations for a complex internal isothermal flow," *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, 190, 2893–2907.

van Kan, J. (1986), "A 2nd-order accurate pressure-correction scheme for viscous incompressible-flow," *SIAM Journal on Scientific and Statistical Computing*, 7(3), 870–891.

Walker, P. and Yoganathan, A. P. (1992), "In vitro pulsatile flow hemodynamics of five mechanical aortic heart valve prostheses," *European Journal of Cardiothoracic Surgery*, 6(Supp 1), S113–S123.

Wang, M. and Moin, P. (2002), "Dynamic wall modelling for large-eddy simulation of complex turbulent flows," *Physics of Fluids*, 14(7), 2043–2051.

Wardhana, K. and Hadipriono, F. C. (2003), "Analysis of recent bridge failures in the United States," *Journal of Performance of Constructed Facilities*, 17(3), 144–150.

Wieting, D., Breznock, E., and Kafesjian, R. (1990), "Exercising sheep: an in vivo model for assessing the durability of pyrolytic carbon heart valves," *International Journal of Artificial Organs*, 13(9).

Wright, J. and Shyy, W. (1993), "A pressure-based composite grid method for Navier-Stokes equations," *Journal Computational Physics*, 107, 225–238.

Wu, W., Rodi, W., and Wenka, T. (2000), "3D numerical modeling of flow and sediment transport in open channels," *Journal of Hydraulic Engineering*, 126(1), 4–15.

Wurzinger, L., Blasberg, P., and Schmid-Schonbein, H. (1985), "Towards a concept of thrombosis in accelerated flow: Rheology, fluid dynamics and biochemistry," *Biorheology*, 22(5), 437–449.

Yao, Y., Savill, A., Sandham, N., and Dawes, W. (2002), "Simulation and modelling of turbulent trailing-edge flow," *Flow Turbulence and Combustion*, 68(4), 313–333.

Yoganathan, A. P., Woo, Y., and Sung, H. (1986), "Turbulent shear stress measurements in the vicinity of aortic heart valve prostheses," *Journal of Biomechanics*, 19, 433–442.

Yuan, L. (2002), "Comparison of implicit multigrid schemes for three-dimensional incompressible flows," *Journal of Computational Physics*, 177(1), 134–155.

Zang, Y., Street, R. L., and Koseff, J. R. (1994), "A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinates," *Journal of Computational Physics*, 114(1), 18–33.

Zhang, X., Schmidt, D., and Perot, B. (2002), "Accuracy and conservation properties of a three-dimensional unstructured staggered mesh scheme for fluid dynamics," *Journal of Computational Physics*, 175(2), 764–791.

Zhao, Y., Tan, H., and Zhang, B. (2002), "A high-resolution characteristics-based implicit dual time-stepping VOF method for free surface flow simulation on unstructured grids," *Journal of Computational Physics*, 183(1), 233–273.

Zhao, Y. and Zhang, B. (2000), "A high-order characteristics upwind FV method for incompressible flow and heat transfer simulation on unstructured grids," *Computer Methods in Applied Mechanics and Engineering*, 190(5-7), 733–756.

Zhou, Z., Todd, B., and Daivis, P. (2003), "Parallelisation of nonequilibrium molecular dynamics code for polymer melts using OpenMP," *Computational Sicence - ICCS 2003, PT III, Proceedings*, 2659, 275–285.